

移动平台典型应用的身份认证问题研究

张效林^{1,2}, 谷大武^{1,2}, 张驰¹

(1. 上海交通大学电子信息与电气工程学院, 上海 200240;

2. 西安电子科技大学网络与信息安全学院, 陕西 西安 710126)

摘要: 近年来的研究表明, 针对USIM卡的攻击手段日益增多, 在5G环境下攻击者也能使用复制的USIM卡绕过某些正常应用的身份认证, 进而获取用户信息。在USIM可被复制的条件下, 研究了移动平台上典型应用的身份认证流程, 通过分析用户登录、重置密码、执行敏感操作的应用行为给出身份认证树。在此基础上, 测试了社交通信、个人健康等7类58款典型应用, 发现有29款认证时仅需USIM卡接收的SMS验证码便可通过认证。针对该问题, 建议应用开启两步验证, 并结合USIM防伪等手段完成认证。

关键词: 移动应用; USIM复制; SMS; 身份认证; 移动应用测试

中图分类号: TP311.1

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020081

Issues of identity verification of typical applications over mobile terminal platform

ZHANG Xiaolin^{1,2}, GU Dawu^{1,2}, ZHANG Chi¹

1. School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

2. School of Cyber Engineering, Xidian University, Xi'an 710126, China

Abstract: Recent studies have shown that attacks against USIM card are increasing, and an attacker can use the cloned USIM card to bypass the identity verification process in some applications and thereby get the unauthorized access. Considering the USIM card being cloned easily even under 5G network, the identity verification process of the popular mobile applications over mobile platform was analyzed. The application behaviors were profiled while users were logging in, resetting password, and performing sensitive operations, thereby the tree model of application authentication was summarized. On this basis, 58 popular applications in 7 categories were tested including social communication, healthcare, etc. It found that 29 of them only need SMS verification codes to get authenticated and obtain permissions. To address this issue, two-step authentication was suggested and USIM anti-counterfeiting was applied to assist the authentication process.

Key words: mobile application, USIM cloning, SMS, authentication, mobile app testing

收稿日期: 2020-02-28; 修回日期: 2020-07-03

通信作者: 谷大武 dwgu@sjtu.edu.cn

基金项目: 国家电网有限公司总部科技项目(2019GW-12)

Foundation Item: Security Protection Technology of Embedded Components and Control Units in Power System Terminal (2019GW-12)

论文引用格式: 张效林, 谷大武, 张驰. 移动平台典型应用的身份认证问题研究[J]. 网络与信息安全学报, 2020, 6(6): 137-151.
ZHANG X L, GU D W, ZHANG C. Issues of Identity Verification of Typical Applications over Mobile Terminal Platform[J]. Chinese Journal of Network and Information Security, 2020, 6(6): 137-151.

1 引言

1.1 研究背景

USIM(全球用户识别模块)^[1]作为 UMTS(通用移动通信系统)网络中用户身份的标识模块广泛用于各类移动设备,为用户提供认证服务、短信息服务等。与 SIM(用户身份识别模块)相比,USIM 卡在应用支持与安全性上进行了升级,USIM 卡在支持 3G/4G 服务的同时,能向下兼容 SIM 卡所支持的 2G 网络。在日渐成熟的 5G 网络中,USIM 卡将在实体认证、信息交换等方面发挥更重要的作用。

移动设备的推广和移动应用的发展相辅相成,互为促进。截至 2019 年第三季度,国家统计局数据显示,我国 4G 移动电话用户数同比增长 10.1%,移动互联网接入流量同比增长 34.9%^[2];同时,5G 基站已投入建成 13 万座,5G 通信技术及商业化迎来高速发展。而在全球范围内,2018 年世界各地用户共计下载了 1 940 亿次手机应用(App),涵盖金融支付类、社交通信类、出行类、娱乐影音类应用等,其中在线金融支付类应用发展迅速,用户下载量较 2017 年相比增长了 27.8%^[3-4],随之而来的数据隐私风险不断增大。搭载 USIM 卡的移动设备及其中的应用程序已成为人们生活的一部分,而这些应用往往拥有用户敏感信息的操作权限,并对登录用户的身份进行认证。在实现身份认证时,SMS 验证码作为一种低成本、易实现、用户学习门槛低的验证方式得到了极为广泛的应用。USIM 卡作为接收 SMS 验证码的必要载体,一旦出现可被复制或伪造等安全问题,将会直接威胁所有 USIM 设备的安全性,也必然会对设备中 App 的身份认证过程与用户隐私产生巨大威胁。

1.2 SIM/USIM 安全研究现状

用户使用移动设备通信时,设备中的 SIM/USIM 卡需先与网络完成认证并接入。研究表明,尽管 USIM 卡使用 MILENAGE 算法实现双向认证,SIM 卡使用 A3/A8 算法实现单向认证,但都面临被复制的可能。

1.2.1 SIM/USIM 卡复制攻击

Liu 等^[5]针对 USIM 认证过程中的功耗信号差

异,提出了一种可行的差分功耗分析(DPA)的侧信道攻击方法,该方法针对 3G/4G AKA 协议^[6]中基于 AES 的 MILENAGE 算法,通过选择其中的 f_5 函数进行差分功耗分析,借助 Person 相关系数^[7]可推算出认证参数运算所用的轮密钥及 OPc 的期望值。基于此,攻击者仅通过使用示波器、智能卡分析仪以及个人计算机即可在若干分钟内完成对原 USIM 卡的复制,实现与 AuC(认证中心)的认证并正常通信。

除此以外,Anwar 等^[8]对基于 COMP128 的 A3/A8 算法进行研究,结合 A3 认证算法的 SRES 响应数对 AuC 所用的伪随机数发生器进行破解,可提取 AuC 与 SIM 卡的客户认证密钥,进而实现了 SIM 卡的复制。Singh 等^[9]考虑到 GSM(全球移动通信系统)所用 COMP128 认证算法可使攻击者成功利用暴力破解提取 SIM 的认证密钥,提出了针对 SIM 卡实施 OTA 复制的基本流程与常见方法。Rao 等^[10]提出了一种称为分区攻击的侧信道攻击方法,该攻击方法能对 COMP128 中划分后的查找表结构实施快速功耗分析,从而提取认证密钥。对于 CDMA 技术,Zhang 等^[11]对 CAVE 协议中的查找表以及 AKA 协议中循环移位操作进行分析,分别设计了不同的功耗分析方法,在 8 位微处理器以及 SIM 卡上花费很短时间即可成功实现认证密钥的提取。

在 5G 网络环境中,USIM 所采取的 AKA 认证协议在 NSA 模式下与 3G/4G AKA 协议的主要流程和算法参数一致^[12]。因此,3G/4G 环境下 USIM 卡的安全分析方法与破解手段在 5G 网络中依然有效^[13-14]。上述研究表明,国内外市场上大量存在和使用着容易被复制的 USIM 卡,攻击者仅需花费几分钟的功耗数据采集时间,即可实现对目标用户 USIM 卡的离线破解和复制,而且这种破解和复制使用个人 PC 只需几分钟到几十分钟的时间即可完成。

为了应对 USIM 卡被上述攻击方法破解的风险,国内外许多芯片设计企业已经开始研究各种芯片防护方法,并将防护技术应用到新设计和流片的 USIM 芯片中。但与软件漏洞的修复相比,芯片安全问题的解决往往需要对芯片进行重新设计开发和流片,这无疑是一个较长的时间,而且

对于 USIM 芯片而言, 即使已经生产出具有抗攻击能力的 USIM 卡, 这些芯片要得到广泛使用, 仍需要运营商对已发行的 USIM 芯片进行大规模的召回或强制更换, 这显然是不可行的。这在客观上造成现实中必然存在大量可被复制的 USIM 卡长期且广泛地被使用。

1.2.2 SMS 验证码的应用及安全

SMS 验证码已经广泛用于登录移动平台上的应用程序或重置密码等认证环节, 作为 USIM 卡向 App 传递信息的载体与桥梁, 它代表了特定 USIM 卡和设备持有人之间的联系。SMS 验证码实质是一种基于时间的一次性密码 (TOTP, time-based one time password)^[15], 其架构如图 1 所示。

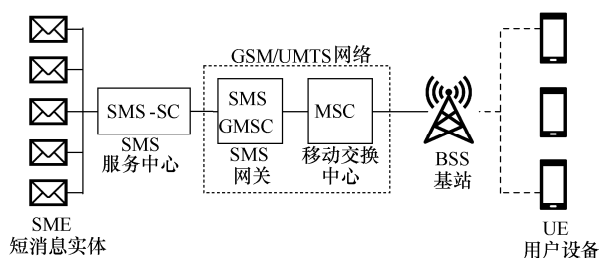


图 1 SMS 一般架构

Figure 1 Overall structure of short message service (SMS)

MSC (移动交换中心) 到 UE 的认证与消息传递往往基于 GSM 或者 UMTS 网络, 实际中针对 SMS 的攻击行为多发生在设备与基站的认证过程中或者用户收到 SMS 消息后。Dmitrienko 等^[16]分析各类 SMS 攻击向量, 指出在设备上安装恶意软件窃取数据是针对 SMS 安全的常见攻击手段。Alecu^[17]则实现了一种无须用户许可就能使设备发送 SMS 消息并让该设备无法收到消息的攻击手段。当攻击者使用复制 USIM 卡接入基站后, 可代替原用户接收 SMS 验证码, 并完成 App 中的认证流程。

1.3 本文的主要工作和结果

为避免 USIM 卡被复制而可能导致的 App 应用的安全漏洞或潜在危害, 目前可行的办法是通过升级 App 或者安全补丁的方式, 从软件层面及时弥补因使用复制的 USIM 所带来的安全风险。在该背景下, 本文通过对典型 App 的详细分析和测试, 指出哪些 App 在 USIM 被复制的情况下是

存在安全问题的, 从而给这些 App 的更新换代和技术升级敲响警钟。

本文针对移动应用身份认证的一般流程总结了认证的一般模型, 重点分析了在复制 USIM 卡的环境中身份认证策略可能存在的安全问题。在模拟 USIM 卡被攻击者复制的环境下, 本文研究了移动终端平台上的 58 款典型应用的身份认证环节, 并进行真机测试, 重点观察应用登录、重置密码、执行敏感操作的认证流程, 对应用的登录数据与跳转流程展开分析, 研究这些应用在身份认证过程中的请求响应数据格式与代码执行过程, 最后记录了各类安全服务 (如 SMS 验证码) 在复制 USIM 卡环境中的使用与表现。

测试结果发现在 58 款应用中有 29 款的身份认证服务在复制 USIM 卡的环境下可被直接绕过。其中有 9 款 App 在密码重置和登录时可被直接绕过, 有 10 款 App 仅在密码重置时可直接被绕过, 剩余 10 款 App 仅在登录时可直接被绕过。本文的测试结果表明, 针对复制 USIM 卡攻击而产生身份认证问题的应用而言, 移动 App 开发者与安全厂商应当至少采用两步认证等软件防护手段才能避免 USIM 卡复制攻击对移动应用带来的安全风险。

2 移动应用的一般认证流程

应用为用户执行功能前需要验证其身份, 而各类 App 提供的身份认证接口通常存在于用户登录、用户重置密码、用户执行敏感操作, 本节分析移动应用身份认证的一般模式, 并总结用户身份的验证树。

2.1 身份认证的功能情景

用户登录认证: 当用户接入应用时, App 需要鉴别并认证该用户身份。用户登录通常要求填写匹配的用户名与口令。而常见的用户名类型通常为用户自定义字符串、用户邮箱、用户手机号等; 口令与用户名匹配后会允许当前用户登录, 一般的用户登录认证流程如图 2 所示。

用户重置密码认证: 实际中用户忘记口令的行为时有发生, 应用也会提供找回密码功能。用户提供正确的用户名后, 需结合其他信息验证用户名的真实性。用户名类型决定了重置密码时认

证身份的流程。使用邮箱地址或手机号作为用户名时,应用会发送验证邮件或短信验证码,只有用户接收到验证信息并执行对应操作后才会允许重置口令,部分应用还会采取两步验证等手段。重置密码认证流程如图 3 所示。

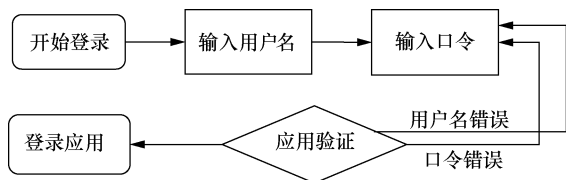


图 2 App 用户登录认证

Figure 2 Procedure of authentication for App users logging in

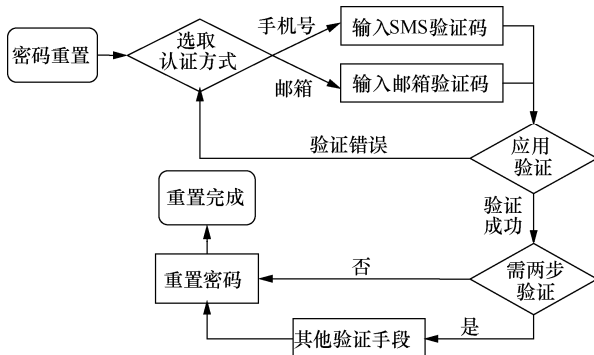


图 3 App 用户重置密码认证流程

Figure 3 Procedure of authentication for App users resetting passwords

电子邮箱与手机作为日常生活的重要组成部分,私人邮箱信件与短信验证码通常为个人所有,因此应用厂商将其作为身份认证的必要信息是合理的,但部分应用未采取额外验证手段确保当前用户身份的正确性,仅保证验证手段的必要性。一旦外部攻击者设法提供正确的验证信息,同样可重置密码,拥有原用户的操作权限,达成攻击目的。

用户执行敏感操作认证:用户登录后若执行某些敏感操作,如转账或查看操作历史,某些应用会要求用户进行额外认证,通常为要求用户输入 PIN 码、短信验证码或生物验证等。用户执行敏感操作认证的流程如图 4 所示。

这一措施可更好地保护业务核心数据,个人金融类 App 涉及银行业务时往往建议用户开启操作认证,保护个人财产安全。

2.2 应用的用户身份验证树

根据应用中 3 种常见身份验证流程,本文可总结移动应用中常见的用户身份验证树,如图 5 所示。

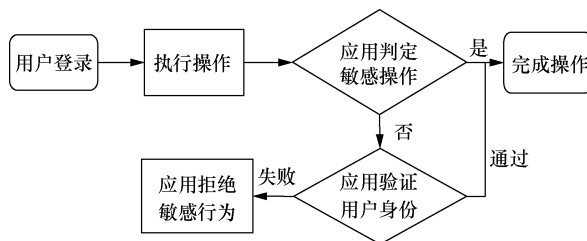


图 4 App 用户执行敏感操作认证流程

Figure 4 Procedure of authentication for App users performing sensitive tasks

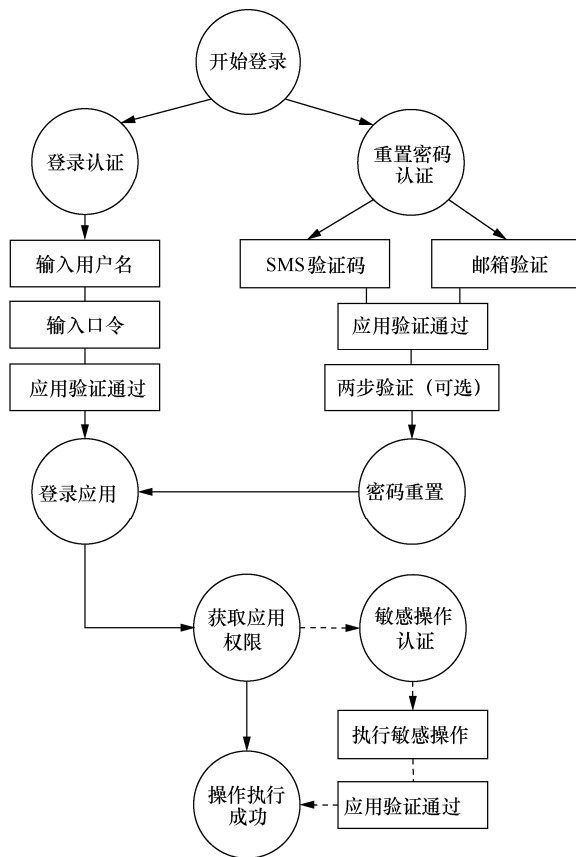


图 5 App 用户身份验证树

Figure 5 Tree model of authentication process in mobile application

在验证过程中,为实现登录应用,攻击者可针对用户登录认证与重置密码认证实施攻击。用户登录认证过程中,可利用应用潜在的逻辑漏洞发起攻击;而在重置密码过程中,所涉及的认证方式较多,这也导致攻击者拥有更多攻击发起点,如针对 USIM 卡的 SMS 验证码安全、邮件安全等攻击手段。同时,由于存在部分应用未开发重置密码时的二次验证流程,攻击者面临的攻击难度进一步降低。如图 5 所示,一旦攻击者成功登录应用,且该应用未对

敏感操作实施再次验证, 则攻击者能够获取应用权限, 窃取用户信息。

3 移动终端平台的应用测试

本节采取实地测试的方法, 考察不同应用在认证过程中的行为表现, 并对移动应用的常见测试类型以及测试技术等进行介绍。

3.1 移动应用的测试分析技术

常见测试类型: 移动应用的各类测试有助于提高软件质量, 以保证软件版本的长期稳定迭代。移动应用的测试方法在测试目标和测试方法上的主要分类结果如图6所示。

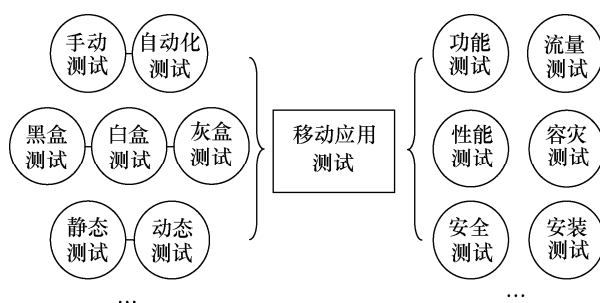


图6 移动应用测试分类
Figure 6 Classification of mobile applications

其中, 功能测试考察应用的基本服务、用户交互以及灵活性。安全测试、流量测试、耗电量测试等近年来成为必要的测试环节^[18]。自动化测试常使用基于黑盒(白盒)的自动化框架, 对产品的模块单元进行动态或静态的分析。为适应产品的快速发展与迭代, 厂商大多采取自动或半自动化的测试方法。

测试分析技术: 文献[19-20]中的 GUI 自动化框架 API 为移动平台的基础系统功能提供了通用接口, 作为其他测试功能的基础。测试人员通过这些 API 编写脚本, 利用断言语句测试状态信息。基于记录重放(R&R)的测试方案如“Reran”^[21]、“Versatile”^[22]等或可替代手动编写测试脚本, 而且这类方法可提供细粒度的捕获和重放。而输入自动生成(AIG)技术^[23]自动化了测试用例的生成过程, 可提高代码覆盖率, 检测更多错误, 减少测试方案规模。此外, 还有错误异常的报告工具^[24]、设备流测试工具等^[25]。

典型应用的行为测试: Wang 等^[26]研究了移

动应用中常见的单次登录(SSO, single sign-on)模型的安全性, 对浏览器中继消息(BRM, browser relayed messages)的字段与传递过程进行黑盒测试与溯源。作者对 Google、Facebook 等应用测试后发现, 攻击者可控制 Google 中的 openid.ext1.required 等字段实现身份冒用。攻击者也能替换 Facebook 中 BRM 的部分字段来欺骗服务端进行授权。Gan 等^[27]则从内容、社交、娱乐的角度对微信与微博用户行为的频率和类型对比分析, 指出微信能给用户提供更高等级的保密以及应用控制。Dmitrienko 等^[28]对 Google Authenticator 等主流验证工具的 OTP 进行了统计分析, 试图从数值分布上寻找 OTP 生成算法的规律, 还指出几种实际中 2FA 验证框架的可能缺陷以及利用手段。

以上研究者对几款典型应用在传统测试环境中的数据交互流程与交互策略进行了分析, 而这些应用均未在复制 USIM 卡的环境中进行测试与研究。本文弥补这一缺陷, 提出在该环境下对各类 App 的行为与流程进行测试, 从而给出这些应用的身份认证中的安全问题和缺陷, 并给出解决思路。

3.2 典型应用的身份认证流程测试

本节在攻击者已拥有复制 USIM 卡环境下, 研究并测试 USIM 设备中 App 的身份认证问题, 通过执行不同 App 的身份认证树, 分析身份认证流程跳转过程中的数据请求与响应结果, 观察并记录 App 的行为, 分析可能被攻击者绕过的环节。

3.2.1 测试条件

测试对象: 应用所含信息价值是影响攻击者选取攻击对象的主要因素之一, 本测试选取更易被攻击者针对的应用, 这些应用通常占据主流市场, 能对大量用户的个人生活产生深刻影响。根据移动数据分析领域著名企业 App Annie^[29]与市场权威调研机构 Statista^[30]的 App 数据统计结果, Business of Apps 与 Buildfire 分别发布了 2019 年 App 下载与使用报告^[3-4], 报告指出, 就 App 下载数、使用程度和卸载数而言, 社交通信类、视频类、外卖类、金融支付类以及游戏类等是移动应用市场中重要的几种应用。结合专业互联网产业研究公司 iResearch 的 App 指数分析^[31]以及国内

表 1 测试应用组成
Table 1 Tested mobile applications

类别	应用组成
社交通信类	WhatsApp, Messenger, Facebook, Instagram, Twitter, Skype, 微信, QQ, SnapChat, Pinterest
金融支付类	Amazon, Wish, Poshmark, eBay, Apple Store, Walmart, Flipkart, AliExpress, SHEIN, 淘宝, 京东, Cash, Paypal, 支付宝, 交通银行网银
出行外卖类	Uber, 滴滴, Lyft, Curb, 12306, Google Maps, 高德地图, 百度地图, Parkmobile, Waze, UberEats, DoorDash, iFood, 美团, 饿了么
健康医疗类	Keep, Nike Training Club, Calm, GoodRx, Pregnancy
文件云盘类	Dropbox, Google Drive, iCloud, Onedrive, 百度网盘
娱乐视频类	Youtube, TikTok, 爱奇艺, Netflix, Amazon Prime Video
信息检索类	Google Chrome, 百度搜索, Bing Search

移动大数据服务商 QuestMobile 发布的 2019 年移动 App 增长报告^[32]可以看出,短视频、综合电商、移动支付等应用发展迅猛。这些应用的用户黏度高,与用户生活结合紧密,可接触到个人隐私等数据,会受到攻击者的研究与针对,而这些应用的潜在安全漏洞也将导致较为严重的数据泄露事件。因此,为提高测试结果的代表性,结合上述 App 分析数据,本文对社交通信类、金融支付类、出行外卖类、健康医疗类、文件云盘类、娱乐视频类和信息检索类共 7 类应用展开测试,每类应用依照下载量与使用量排名选取共 58 款典型应用,详情如表 1 所示。

以上 58 款应用能接触到用户的通信交流内容、财产状况、地理位置与出行轨迹、生理健康状况、私人文件以及检索信息。这些信息与用户隐私直接相关,这些信息的泄露会危害用户数据安全,造成严重的个人信息安全风险,一旦这些应用无法保证用户身份验证的安全性,则会对用户生活造成较大的潜在威胁。

测试环境: 本文所采取的测试机型信息如表 2 所示。

Apple iOS 与 Google Android 为移动设备系统的主要类型,除 iCloud 所代表的 Apple 验证机制,其余 57 款应用在 iOS 与 Android 上具备相同的认证流程。因此,本次测试所用设备对 58 款应用的测试结果与结论符合不同测试系统下的测试结果。基于以上测试条件,在测试前需给出以下前提。

表 2 测试机型信息
Table 2 Tested mobile platforms

指标	信息
机型	iPhone XR
系统	iOS 12.4.1
运行商	中国联通
IMEI	357394092794037
ICCID	89860116208410304191
MEID	35739409279403

1) 复制后的 USIM 卡与原 USIM 卡在收发消息时对基站而言是相同的。

2) 目标用户的用户名与 USIM 卡的手机号是易获取的。

根据如上测试条件,可提出测试的一般方法。

3.2.2 测试方法

本节根据移动应用的验证树进行真机测试,下载各应用后使用目标手机号或邮箱进行注册,完成新用户的一般注册过程。之后退出登录模拟攻击者行为,在登录页面测试重置密码功能,关注不同步骤的响应结果及需要 SMS 验证码的环节,记录应用在密码重置及其他环节是否可绕过。在测试过程中使用 Stream 抓取 App 登录过程中的 Http/Https 数据请求与响应,分析数据包字段与控制方法,结合 App 实际行为表现给出测试结果。测试基本流程如图 7 所示。

其他测试要求如下。

1) 对于可使用手机号或邮箱注册登录的应用而言,优先使用手机注册登录。

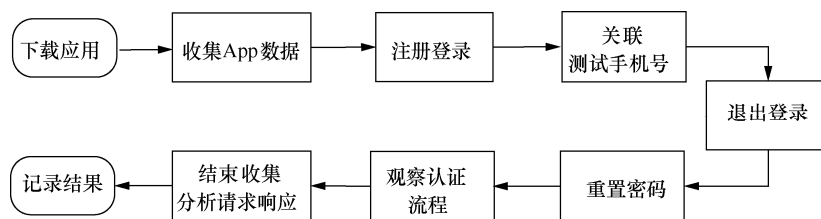


图7 一般测试流程

Figure 7 General procedure of authentication test

2) 对于使用 SSO 服务的应用而言, 其认证逻辑与身份提供方 (IdP, identity provider) 相同, 优先选择不使用 SSO 服务进行测试。

3) 对于默认不开启二次验证的应用而言, 保持原始设置进行测试。若应用强制开启两步验证, 测试时不关闭该功能。

本文选取“是否需要短信验证码”“是否需要密保信息”“是否需要邮箱验证”“是否对当前设备进行验证”作为密码重置过程中的测试指标。结合测试过程中应用的 Http/Https 数据字段以及页面代码中的控制函数, 如果用户仅需短信验证码且无须设备环境验证就能完成密码重置, 则可认为攻击者在复制 USIM 卡环境下是可以直接绕过应用的身份验证环节, 即该应用是不安全的。

3.3 典型应用身份认证中的数据分析

在测试过程中, 本文对应用的请求数据与响应结果进行捕获和分析, 本节以微信与支付宝这两款典型应用为例, 分析其重置密码过程中请求与响应数据的格式与跳转流程, 并结合应用行为, 给出二者在复制 USIM 环境下的测试结果。

3.3.1 微信登录中的数据捕获与分析

根据 3.2.2 节中的要求, 本节需测试微信 (WeChat) 找回密码时认证流程, 由于微信可直接使用手机号码与短信验证码完成登录, 因此在用户选择“找回密码”功能时, 微信会令用户尝试使用手机号直接登录。在此过程中, 利用 Stream

1.0.4 捕获并分析 Http 网络流量。微信登录过程中的主要请求数据如表 3 所示。

对服务端响应数据解析后可知, 以上 3 个请求数据依次对应客户端找回密码时“请求找回密码”“请求手机验证码登录”“请求登录应用”的 3 个阶段, 以下结合请求中的关键字段与响应页面的代码执行过程对微信登录阶段的身份认证流程展开说明。

请求找回密码: 当用户在微信登录页面申请找回密码时, 客户端会向 support.weixin.qq.com 发起请求, 通过 GET 方法将查询字段 t、lang、rid 传递给服务器验证身份, 3 个字段分别表明要请求页面的 URL、客户端语言类型以及客户端身份标识。验证通过后, 服务器进行响应, 客户端跳转至找回密码的提示信息页面。此时微信会提示用户进行手机登录。响应数据中包括该页面的控制代码, 选取“能接收短信”的功能按钮, 对应代码中的 ican 函数将实现下一页面的跳转, 如图 8 所示。在下一页面中确认后, 则由 go 函数构造手机验证码登录应用的请求。

请求手机验证码登录: 当用户可用短信验证码登录时, 根据提示将跳转至使用 SMS 验证码登录页面, 页面前端代码如图 9 所示。其中, go 函数将 p1 与 p2 字段置 1, 并将 rid 赋给 p10 字段, 以上字段拼接并作为 reportData 数据, 该数据作为登录请求的一部分在下一阶段传递

表3 登录微信时的部分 Https 数据请求
Table 3 Extracted data of Https request when logging into WeChat

请求域名	传递方法	上/下行流量 (字节数)	请求字段
support.weixin.qq.com	GET	815 / 22.3×10 ³	t, lang, rid
support.weixin.qq.com	GET	1.61×10 ³ / 335	ap_msg
support.weixin.qq.com	GET	648 / 221	id, c, p10, p0, p1, p2

```
// 微信找回密码界面的部分html语句
1. <div class ="text_area" >
2.   <h1 class ="title" > 你的手机号<em class ="js_mobile" > </em> 目前能不能接收短信?
   </h1>
3. </div>
4. <div class ="opr_area" >
5.   <p class ="btn_area btn_area_inline" >
6.     <a href ="javascript.next()" class ="btn btn_default" >不能</a>
7.     <a href ="javascript.ican()" class ="btn btn_primary" >能接收短信</a>
8.   </p>
9. </div>
10. <script type ="text/html" > reportData+="& p0=1"; </script>
    /* some code */

// ican函数内容
1. function ican(opt){
2.   var ele = document.getElementById(steps[current]);
3.   hideEle(ele.querySelector (".question" ));
4.   showEle(ele.querySelector (".deal" ));
5.   if (opt == "email" ){
6.     //
7.   }
8. }
```

图 8 找回密码界面的部分 html 语句与 ican 函数

Figure 8 An html and javascript code segment for retrieving password and ican function

```
// 微信确认手机登录页面的部分 html
1. <div class ="deal dn" >
2.   <h1 class ="title" >你可以通过 <em>手机号+短信验证码 </em> 登录微信 </h1>
3.   <p class ="btn_area" >
4.     <a href ="javascript.go('&p2=1')" class ="btn" >用短信验证码登录 </a>
5.   </p>
6.   <script type ="text/html" >reportData+="& p1=1"; </script>
7. </div>

// 部分 go 函数及 reportFunc 函数 /* some code */
1. function go(report){
2.   if (report) reportData += report;
3.   reportFunc();
4.   setTimeout( function(){
5.     if (href) location.href = href;
6.   }, 300);
7. }
8. function reportFunc(){
9.   var img = new Image ();
10.  img.src = "https://support.weixin.qq.com/cgi-bin/mmsupport-bin/
    webreport?id=11940&c=11" + reportData;
11. }
```

图 9 确认请求的部分 html 语句和 go 函数

Figure 9 An html and javascript code segment for confirming requests and go function

给服务器。而在此阶段，客户端将会构造一 `ap_msg` 字段，包含当前设备的网络环境与系统参数等信息，并以 GET 方法请求服务端发送验证码登录页面。

请求登录应用：当用户使用短信验证码登录时，客户端认证通过后，`reportFunc` 函数利用 Javascript 的 `Image` 对象的特性，通过更改对象的源链接属性实现对服务器资源的静态访问。最终，消息中的 `reportData` 字段

可告知服务端用户的身份以及使用何种方式登录应用，用 GET 方法发送给服务端完成登录，该部分的 Https 数据请求与响应如图 10 所示。

综上，在微信找回密码与登录过程中，客户端通过 `scan` 与 `go` 等函数实现了根据用户已有身份凭证完成应用身份认证的过程。在客户端，当攻击者在重置密码环节选取用短信验证码登录应用时，可使用复制 USIM 卡直接获取短信验证码完成登录。

```
GET
/cgi-bin/mmsupportbin/webreport?id=11940&c=11&p10=c176add34fa841321a99f72774e1b87b_1585720693&p0=1
&p1=1&p2=1 HTTP/1.1
Host: support.weixin.qq.com
Connection: keep-alive
Accept: image/png,image/svg+xml,image/*;q=0.8,video/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 13_3 like Mac OS X) AppleWebKit/605.1.15
(KHTML, like Gecko) Mobile/15E148 MicroMessenger/7.0.11(0x17000b21)NetType/WIFI Language/zh_CN
Accept-Language: zh-cn
Referer: https://support.weixin.qq.com/security/readtemplate?t=page/login_optimized__w_index&lang=zh_CN&rid=c176add34fa841321a99f72774e1b87b_1585720693
Accept-Encoding: gzip, deflate, br
```

```
HTTP/1.1 200 OK
Content-Type: text/javascript
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache, must-revalidate
Connection: keep-alive
Content-Length: 45
window.reportret =0; window.reportmsg ="all ok"
```

图 10 请求登录应用的 Https 数据请求（上）与响应（下）

Figure 10 Captured raw data of Https request (top) and response (bottom) when successfully logging into WeChat

3.3.2 支付宝重置密码中的数据分析

使用与 3.3.1 节中相同的方法分析支付宝 (AliPay) 的重置密码过程分析。在登录支付宝账号时，本文选取重置密码而非 SMS 验证码登录。使用 Stream 捕获支付宝找回密码中的 Https 数据，主要请求如表 4 所示。

以上请求数据包分别对应客户端请求找回密码、发送 SMS 验证码、发送重置密码。而在支付宝重置密码时，客户端会向 loggw.alipay.com 的主机反复发送包含有手机型号、网络环境、运营商类型等设备环境参数的数据，并由服务端返回的状态码判定当前设备是否能继续下一步操作。由于该环节 Https 流量数据较多，下面仅对表 4 中的 3 个请求响应过程展开说明。

请求找回密码：当使用找回密码功能时，客户端以 GET 方法将用户服务类型、临时 Token、应用版本等字段传递给域名为 clientsc.alipay.com 的服务器。服务器端则会返回包含页面控制代码的响应页面供用户选择找回密码的方式。与微信类似，支付宝也是通过 javascript 代码来控制用户

找回密码的身份认证流程，如图 11 所示。用户输入手机号并确认可收到 SMS 验证码后，会由 anima-yocto 触发 canGetAckCode 并告知用户验证码已发送。

当用户获知服务端发送验证码后，客户端将显示 p_validateMobile 页面提示用户输入验证码，并由支付宝客户端构造新的请求数据，如图 12 所示。

接收并输入 SMS 验证码：当用户收到验证码后，在 p_validateMobile 页可以输入 4 位短信验证码。与微信不同之处在于支付宝的 SMS 验证码值将被包含在请求数据中由服务端进行检查。故此处客户端调用 sendAckCode 方法完成传输预认证，再调用 checkAckCode 方法将输入的验证码作为 ackcode 字段值，并以明文 POST 的方法向 clientsc.alipay.com 主机中的 checkAckCode.json 资源发起请求，由服务器检查 ackcode 的正确性。当验证码输入正确，服务端的响应字段 stat 被置为“ok”后，客户端由 checkAckCode 方法请求服务端发送重置密码页面。

表 4 支付宝找回密码时的部分 Https 数据请求
Table 4 Extracted data of Https request when resetting password in Alipay

请求域名	传递方法	上/下行流量（千字节数）	请求字段
clientsc.alipay.com	GET	1.21 / 19.0	Serveid、authorizeToken、donotCloseH5、walletVersion、autologin、callback
clientsc.alipay.com	POST	1.28 / 0.543	ackCode、bizTokenForSecurity
clientsc.alipay.com	POST	2.55 / 0.810	newPassword、bizTokenForSecurity、envData

```
// 支付宝使用手机号码找回密码页面
1. <div class="am-ft-lg am-ft-gray">您的手机号
2. <span class="J-country-code am-ft-black"></span>
3. <span class="am-ft-20 J-mobile-phone ft-strong-l p-tl am-ft-black">
4. 132*****68</span> 现在能否接收短信?
5. </div>
6. <div class="am-button-group" role="group" aria-label="操作按钮组">
7. <button type="button" id="canNotGetAckCode" class="am-button white">不能</button>
8. <button type="button" id="canGetAckCode" class="am-button blue">能</button>
9. </div>
// yocto触发canGetAckCode
1. yocto('#canGetAckCode').on('tap', function() {
2. yocto('.p_validateMobileGuide').addClass('fn-hide');
3. yocto('.p_validateMobile').removeClass('fn-hide');
4. if (!ackCodeInstance) {
5. // set ackCodeInstance
6. } else {
7. // send ack code
8. }
9.});
```

图 11 找回密码提示页面的部分 html 语句和 yocto 触发操作
Figure 11 A code segment for retrieving password and yocto operation

```
POST /account/checkAckcode.json HTTP/1.1
Host: clientsc.alipay.com
Accept: */*
Accept-Language: zh-CN,en-US;q=0.8
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 13_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like
Gecko) Mobile/17C54 NebulaSDK/1.8.100112 Nebula WK PSDType(1)
AlipayDefined (nt:WIFI,ws:414|832|2.0) AliApp(AP/10.1.90.8000) AlipayClient /10.1.90.8000
Language/ zh-Hans Region/CN
Content-Length: 100
Connection: keep-alive
ackCode=7730&bizTokenForSecurity=wallet_20010%7C5ae275c6-919c-4901-b19e-e6b7d7b85850%7C15857223
78598
```

```
HTTP/1.1 200 OK
Server: Tengine/2.1.0
Content-Type: application/json;charset=GB18030
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=31536000
Content-Language: zh-CN
Set-Cookie: spanner=BuEH+zeKzqn mYTJGX4QBolhBGRnSlynU;path=/;secure;
Via: spanner - internet - 5318.et15 [200]
{"stat":"ok","loginId":""}
```

图 12 客户端发送验证码的 HTTPS 数据请求 (上) 与响应 (下)
Figure 12 Captured raw data of https request (top) and response (bottom) when send authentication code in Alipay client

发送重置密码：重置密码页的前端代码如图 13 所示。当客户端提示用户输入新密码时，由 yocto 捕捉页面的 newPassword 字段，并触发 saveNewQueryPwdBtnClickCb 与 resetQueryPwd 方法实现新密码的重置与发送。后者将用户新密码封装到请求数据中的 newPassword 字段，并以 POST 方法请求 clientsc.alipay.com 中的 resetQueryPwd.json 资源确认新密码。当服务端的响应字段 stat 被置为“ok”后，客户端将跳

转至支付宝应用界面，完成密码重置。

综上，在客户端发送收到的 SMS 验证码以及重置密码过程中，支付宝客户端均先请求特定的安全服务包括 Google 的 URL 检查服务 safebrowsing 和 loggw.alipay.com 中的设备环境验证服务。因此，在复制 USIM 卡环境下，攻击者无法绕过服务端对用户终端设备参数环境的检查与验证环节，则仅使用手机号码与 SMS 验证码实现口令重置是困难的。

```
// 重置密码的resetQueryPwd 函数
1. function resetQueryPwd() {
2.   yocto.ajax({
3.     type: 'POST',
4.     url: '/account/resetQueryPwd.json',
5.     timeout: 10000,
6.     data: {
7.       newPassword: alipay.security.products['newPassword'].getRsaPassword(),
8.       bizTokenForSecurity: yocto('#sp').val(),
9.       envData: envData
10.    },
11.    success: function (data) {
12.      if (data.stat === 'ok') { // success
13.        // fail
14.      }
15.    }
16.  }
17. }

// 响应主体数据
{
  "authorizeToken": " ",
  "wallet_200107f34f018-b7bc-4dde-9d9f-acc292439ff41585722377891",
  "smsAuthFlag": true,
  "resultMsg": "您的淘宝账号及支付宝账号都只能使用这个密码进行登录，切记",
  "stat": "ok",
  "callback": "https://www.alipay.com/webviewbridge",
  "loginId": "",
  "showType": "confirm"
}
```

图 13 resetQueryPwd 的封装部分与服务器响应主体
Figure 13 A glance of resetQueryPwd function and response body

4 典型应用的身份认证测试结果与分析

根据第 3 节中的测试要求与测试方法，通过对测试应用的登录与找回密码等身份认证功能的流程观察与记录，本节给出测试结果与分析。

4.1 测试结果概述

根据记录，在重置密码时，测试用的 58 款应用有 19 款可直接通过短信验证码重置密码完成登录，其余 39 款由于额外要求的密保信息、邮箱验证以及设备验证等手段使攻击者无法绕过。

如图 14 所示，在这 39 款应用中，有 18 款不可被绕过是因为该应用仅提供邮箱验证而未提供短信验证功能。除 Apple 旗下的 App store 以及 icloud，其他测试应用的验证手段均为短信验证或邮箱验证。在 21 款执行二次验证的应用中 4 款采取了额外的密保信息，4 款采取了额外的邮箱验证，13 款应用会对当前设备环境进行检测，提醒或警告原登录用户当前攻击者的行为，防止攻击者直接绕过。这 21 款应用中部分应用同时借助了多种验证手段确保产品安全。密码重置时无法被绕过的 39 款应用测试指标关系如图 15 所示。

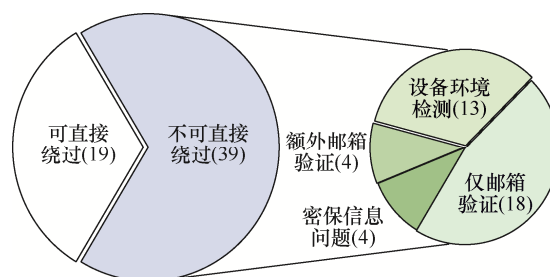


图 14 测试结果组成
Figure 14 Overall experimental results

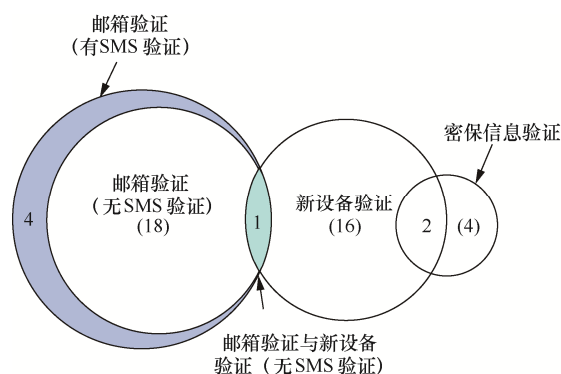


图 15 不可绕过应用的测试项目关系
Figure 15 Relations of Apps whose authentication process can not be bypassed

测试还发现，当攻击者可获取发往用户设备的 SMS 验证码时，有 9 款应用在密码重置和正常

登录时均可被绕过,有 10 款应用仅在密码重置时易被绕过,有另外 10 款应用仅在正常登录时可被绕过,如图 16 所示。

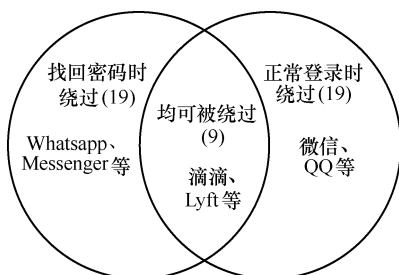


图 16 被绕过结果关系

Figure 16 Relations of Apps whose authentication process can be directly bypassed

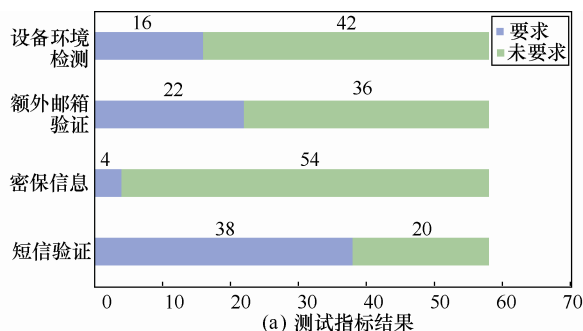
其中,各指标所包含应用名称如表 5 所示。

当额外采取至少一种验证手段时,攻击者很难窃取用户信息。但正常登录时攻击者可利用用户手机号进行登录,这使各应用默认当前异地登录者为原合法用户,从而不进行额外验证。

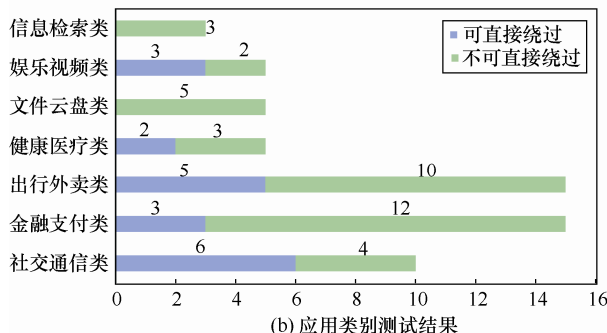
4.2 应用分类与指标测试结果

对于各测试指标而言,有 38 款应用提供了短信验证重置密码的接口,仅有 4 款应用要求用户提供密保信息,如历史订单、购买过的产品名称等。对于邮箱验证,有 4 款应用在要求短信验证的情况下仍需要额外的邮箱验证,而有 22 款则仅需要邮箱验证,还有 16 款应用会进行新登录设备的环境检测等。

如图 17(b)所示,对于测试的 7 类应用而言,金融类应用被直接绕过的比例最少,仅为 20%。而在社交类、健康类以及娱乐类共 20 款应用中,有 11 款可被绕过。在全球移动应用市场中,直接涉及用户个人财产安全的应用具有相对安全的保护措施,而提供娱乐消遣以及个人健康信息等间接涉及用户隐私的应用仍具有较简单的身份认证策略,易被攻击者绕过。



(a) 测试指标结果



(b) 应用类别测试结果

图 17 应用测试指标结果

Figure 17 Detailed results under different authentication requirements and App categories

4.3 测试结果分析与研究

本节结合测试结果,从应用流程的特点、应用类型与认证流程的关系等角度对移动应用的身份认证流程现状以及特点进行分析。

4.3.1 国内外 SMS 验证服务的差异

在测试过程中注意到大多数中国区应用实现了该服务入口,而较多国外应用仅使用邮箱及用户名进行认证。进一步分析发现,国内的 17 款应用全部提供了短信验证重置密码服务,有 6 款被认定为可直接绕过,而国外应用中有 21 款提供了短信验证,13 款可被绕过,如图 18 所示。

移动互联网企业与相关部门大规模推行 SMS 验证码服务的原因不仅在于短信验证码的成本低、操作难度小,更在于实现对移动应用的有效监管。测试还发现,国外的 App 并未强制要求

表 5 可绕过应用组成
Table 5 Composition of flawed Apps

指标	数量	应用名称
密码重置时可被直接绕过	19	WhatsApp、Messenger、Facebook、Instagram、Twitter、SnapChat、Flipkart、Cash、Paypal、滴滴、Lyft、高德地图、iFood、饿了么、Keep、Nike Training Club、Tiktok、爱奇艺、Netflix
正常登录时可被直接绕过	19	Wechat、QQ、Alipay、淘宝、京东、Cash、滴滴、Lyft、高德地图、百度地图、iFood、美团、饿了么、Keep、GoodRx、百度网盘、Tiktok、爱奇艺、百度
密码重置与正常登录均可被绕过	9	Cash、滴滴、Lyft、高德地图、iFood、饿了么、Keep、Tiktok、爱奇艺

绑定手机号, 甚至少部分 App 无手机号注册入口。

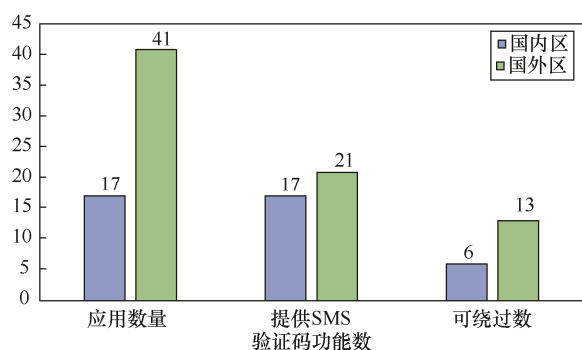


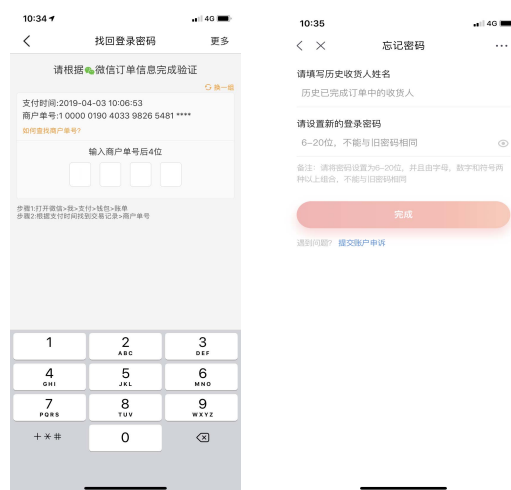
图 18 国内外应用对比

Figure 18 Comparison between domestic and foreign Apps

4.3.2 密保信息的缺失

对于各指标的测试结果而言, 仅有 6.9% 的应用使用密保信息作为第二步验证手段。对于不同类型的密保信息而言, 独有性过高可能增大用户的操作难度, 而独有性较低则会在各类社会工程学手段面前显得十分脆弱。

图 19(a)表明应用所要求的密保信息为用户以往订单中随机一条商家单号的后四位, 该操作对于不经常使用移动智能手机的用户而言有一定难度。而图 19(b)应用则要求填写历史订单中任意一位收货人姓名, 而该信息易被各种钓鱼及检索手段泄露。因此, 用户操作难度低, 但独有性高的密保信息的设计应成为未来 App 用户身份验证体系应用中需要考虑的问题。



(a) 密保信息为商户单号 (b) 密保信息为用户姓名

图 19 两测试用 App 密保信息界面

Figure 19 Front page of "Password Protection Information Query" in two testing Apps

4.3.3 应用信息类型的价值差异

对于不同种类应用而言, 具有较低的绕过比例仅说明在复制 USIM 卡环境下有较好的安全表现, 但无法说明其在保护用户信息上具有较强的防护程度。较多零售类以及少部分支付类应用在注册时并不需要额外的身份绑定, 而且二次验证功能非强制开启。可以认为, 在用户和移动互联网企业的信息价值评价体系中, 与个人在线资产(如所有财产、云文件等)相比, 个人兴趣爱好、健康状况以及社交内容体现出的敏感程度不高, 使娱乐、健康医疗、社交这类应用所拥有较单一的验证手段, 从而导致这 3 类应用有较高的可绕过比例。

4.3.4 设备验证手段的完善

测试发现, 重置密码时有 27.6% 的应用会对用户的设备型号、所处环境进行检测, 如对当前网络 IP 地址、地理位置、机器的硬件参数等进行抓取, 通过与原设备常用登录信息比对等识别当前试图登录的用户或提醒原设备用户。

目前对新设备验证的手段主要有短信验证、设备检测及环境检测这 3 种。

设备检测环节则是 Google 等应用检测到新设备登录时采取的两步验证手段之一。当攻击者试图在某设备重置密码时, 原设备登录的应用会提示用户是否允许该操作的发生。只有用户在原设备账户中完成认证, 才能实现其他设备重置密码的请求。例如, 当攻击者试图在某设备对目标 Apple ID 进行重置密码操作时, Apple 要求该 ID 登录其他设备先行验证。此举可以有效防止攻击者利用复制的 USIM 卡登录应用。

5 结束语

本文主要研究移动应用的身份认证问题, 首先介绍移动应用的认证流程并总结移动应用的一般验证树, 以指明身份认证策略中存在问题的环节; 然后在各类移动应用认证用户身份时, 登录应用、重置密码以及执行敏感操作涉及的认证方式可能不同, 而 SMS 验证码以及邮件验证等方式则广泛应用于重置密码以及登录认证的过程。USIM 卡复制攻击使攻击者获取发往原用户设备的 SMS 验证码成为可能, 从而能以验证树中重置

密码以及应用登录为攻击起点, 绕过认证登录应用后, 窃取用户信息。

本文采用真机测试, 模拟攻击者成功实施 USIM 卡复制攻击后各类应用的密码重置与登录认证的安全缺陷。分析身份认证过程中的 Https 流量数据并结合应用行为后发现, 在 58 款应用中 共有 29 款面临可被复制 USIM 卡直接绕过的安全风险, 其中有 9 款应用在密码重置和登录时可被直接绕过, 有 10 款应用仅在密码重置时可直接被绕过, 剩余 10 款应用仅在登录时可直接被绕过。其中, 社交通信类以及娱乐类应用最易被绕过。移动应用开发者与安全人员应对当前产品的认证逻辑进行完善与改进, 采用两步认证或双因子认证等手段防御 USIM 卡复制攻击, 实现用户数据安全防护手段的有效部署。

致谢:

本文工作得到了上海交通大学蚂蚁金服安全技术联合研究中心和上海交通大学智巡密码安全测评联合实验室的支持, 特此致谢。

参考文献:

- [1] 3GPP specification:31.102. Characteristics of the Universal Subscriber Identity Module (USIM) application[S].
- [2] 国家统计局. 电信业务统计数据[EB].
- [3] DOGTIEV A. App download and usage statistics (2018), business of App (2018)[EB].
- [4] BLAIR I. Mobile App download and usage statistics, BuildFire. [EB].
- [5] LIU J R, YU Y, STANDAERT F, et al. Small tweaks do not help: differential power analysis of milenage implementations in 3G/4G USIM cards[C]//The 20th European Symposium on Computer Security. 2015: 468-480.
- [6] BRIER E, CLAVIER C, OLIVIER F. Correlation power analysis with a leakage model[C]//Cryptographic Hardware and Embedded Systems. 2014: 16-29.
- [7] 3GPP specification: 35.206. Specification of the MILENAGE algorithm set[S].
- [8] ANWAR N, RIADI I, LUTHFI A. Analisis SIM card cloning terhadap algoritma random number Generator[J]. Buana Inform, 2016, 7(2): 143-150.
- [9] SINGH J, RUHL R, LINDSKOG D, et al. GSM OTA SIM cloning attack and cloning resistance in EAP-SIM and USIM[C]//Proc Soc 2013: 1005-1010.
- [10] RAO J R, ROHATGI P, SCHERZER H, et al. Partitioning attacks: or how to rapidly clone some GSM cards[C]//Proc IEEE Symp Secur Priv, 2002: 31-41.
- [11] ZHANG C, LIU J R, GU D W, et al. Side-channel analysis for the authentication protocols of CDMA cellular networks[J]. J Comput Sci Technol, 2019: 1079-1095.
- [12] 3GPP specification: 33.501. Security architecture and procedures for 5G System[S].
- [13] BASIN D, RADOMIROVIC S, DREIER J, et al. A formal analysis of 5G authentication[C]//Proceedings of the ACM Conference on Computer and Communications Security. 2018: 1383-1396.
- [14] LoCCS GoCE. 抱紧你的 SIM 卡—5G 物理安全初探[EB]. LoCCS GoCE, hold your SIM card tight—a glance at 5G physical security[EB].
- [15] KOOT L. Security of mobile TAN on smartphones a risk analysis for the iOS and Android smartphone platforms[D]. The Netherlands: Radboud University Nijmegen, 2012.
- [16] DMITRIENKO A, LIEBCHEN C, ROSSOW C, et al. Security analysis of mobile two-factor authentication schemes[J]. Intel Technol J, 2014: 138-161.
- [17] ALECU B. SMS Fuzzing-SIM Toolkit attack[R]. 2013.
- [18] KIM H K, YEO H, HWANG H J, et al. Effective mobile applications testing strategies[R]. 2016.
- [19] Google. Espresso testing android-framework, Github[EB].
- [20] Apple. Apple ui-automation documentation, Apple[EB].
- [21] GOMEZ L, NEAMTIU I, AZIM T, et al. Reran: timing-and touch-sensitive record and replay for android[C]//35th International Conference on Software Engineering (ICSE). 2013: 72-81.
- [22] HU Y, AZIM T, NEAMTIU I. Versatile yet lightweight record-and-replay for android[C]//Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications. 2015: 349-366.
- [23] MACHIRY A, TAHILIANI R, NAIK M. Dynodroid: an input generation system for android Apps[C]//Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. 2013: 224-234.
- [24] Appsee. App mobile analytics platform[EB].
- [25] OpenSTF. Smartphone test farm, Github[EB].
- [26] WANG R, CHEN S, WANG X F. Signing me onto your accounts through facebook and google: a traffic-guided security study of commercially deployed single-sign-on web services[C]//2012 IEEE Symposium on Security and Privacy. 2012: 365-379.
- [27] GAN C, WANG W. Uses and gratifications of social media: a comparison of microblog and WeChat[J]. Journal of Systems and Information Technology, 2015(4): 351-363.
- [28] DMITRIENKO A, LIEBCHEN C, ROSSOW C, et al. On the (in) security of mobile two-factor authentication[C]//International Conference on Financial Cryptography and Data Security. 2014: 365-383.

[29] AppAnnie. The App analytics and App industry standard[EB].

[30] Statista. Global business data platform[EB].

[31] iresearch. App 指数分析[EB].

Iresearch. App index Analysis[EB].

[32] QuestMobile. 2019 移动 App 半年增长报告[EB].

QuestMobile. Mobile App semi-annual growth report of 2019[EB].

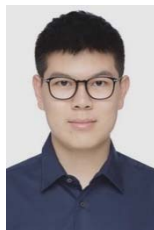


谷大武（1970—），男，河南漯河人，博士，上海交通大学教授、博士生导师，主要研究方向为密码学与系统安全。

[作者简介]



张效林（1998—），男，山西临汾人，上海交通大学硕士生，主要研究方向为信息安全与密码学。



张驰（1991—），男，山西太原人，上海交通大学博士生，主要研究方向为硬件安全、基于机器学习的侧信道分析。