面向 OAuth2.0 授权服务 API 的账号劫持攻击威胁检测

刘奇旭[1,2]，邱凯丽[1,2]，王乙文[1,2]，陈艳辉[1,2]，陈浪平[1,2]，刘潮歌[1,2]
（1. 中国科学院信息工程研究所，北京 100093；2. 中国科学院大学网络空间安全学院，北京 100049）

摘 要：OAuth2.0 授权协议在简化用户登录第三方应用的同时，也存在泄露用户隐私数据的风险，甚至引发用户账号被攻击劫持。通过分析 OAuth2.0 协议的脆弱点，构建了围绕授权码的账号劫持攻击模型，提出了基于差异流量分析的脆弱性应用程序编程接口（API）识别方法和基于授权认证网络流量监测的账号劫持攻击验证方法，设计并实现了面向 OAuth2.0 授权服务 API 的账号劫持攻击威胁检测框架 OScan。通过对 Alexa 排名前 10 000 的网站中真实部署的 3 853 个授权服务 API 进行大规模测试，发现 360 个存在脆弱性的 API。经过进一步验证，发现了 80 个网站存在账号劫持攻击威胁。相较类似工具，OScan 在覆盖身份提供方（IdP）全面性、检测依赖方（RP）数量和威胁检测完整性等方面均具有明显的优势。

关键词：OAuth2.0 协议；应用程序编程接口；账号劫持；第三方应用
中图分类号：TP309.5
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019144

Account hijacking threat attack detection for OAuth2.0 authorization API

LIU Qixu[1,2], QIU Kaili[1,2], WANG Yiwen[1,2], CHEN Yanhui[1,2], CHEN Langping[1,2], LIU Chaoge[1,2]
1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: OAuth2.0 protocol has been widely adopted to simplify user login to third-party applications, at the same time, existing risk of leaking user privacy data, what even worse, causing user accounts to be hijacked. An account hijacking attack model around authorization code was built by analyzing the vulnerabilities of the OAuth2.0 protocol. A vulnerable API identification method based on differential traffic analysis and an account hijacking verification method based on authorized authentication traffic monitoring was proposed. An account hijacking attack threat detection framework OScan for OAuth2.0 authorization API was designed and implemented. Through a large-scale detection of the 3 853 authorization APIs deployed on the Alexa top 10 000 websites, 360 vulnerable APIs were discovered. The further verification showed that 80 websites were found to have threat of account hijacking attack. Compared with similar tools, OScan has significant advantages in covering the number of identity provider, the number of detected relying party, as well as the integrity of risk detection.

Key words: OAuth2.0 protocol, application programming interface, account hijacking, the third-party application

收稿日期：2019−03−22；**修回日期**：2019−05−22
通信作者：刘潮歌，liuchaoge@iie.ac.cn
基金项目：国家重点研发计划基金资助项目（No.2016YFB0801604, No.2016QY08D1602）；中国科学院网络测评技术重点实验室基金资助项目；网络安全防护技术北京市重点实验室基金资助项目
Foundation Items: The National Key Research and Development Program of China (No.2016YFB0801604, No.2016QY08D1602), Key Laboratory of Network Assessment Technology at Chinese Academy of Sciences Program, Beijing Key Laboratory of Network Security and Protection Technology Program

1　引言

近年来，互联网规模持续增长，各类网络应用已经融入现实生活的方方面面。然而为了登录网络应用，用户不得不持有众多账号，并且限于密码策略和“撞库”风险，这些账号的口令往往还略有不同，这给用户带来了相当大的精神负担，严重影响了用户体验。为解决这一问题，谷歌等公司相继提供了跨应用的用户数据共享接口，单点登录（SSO, single-sign-on）便是其中最受欢迎的解决方案之一。

开放授权（OAuth，open authorization）是 SSO 已实现的最成熟的授权认证方法，被广泛地用于简化第三方应用程序的用户身份验证和服务器授权。在本文语境中，“OAuth”既表示授权认证方法，也表示实现这种授权认证的协议。OAuth 协议先后经历了 OAuth1.0、OAuth1.0a 和 OAuth2.0 共 3 个版本。OAuth1.0a 是在 OAuth1.0 基础上的安全升级，但是 OAuth1.0a 过于复杂且易用性差，因而被后来的 OAuth2.0 迅速取代。

各大互联网服务商普遍提供了基于 OAuth2.0 协议的登录授权应用程序编程接口（API, application programming interface）供第三方调用。这样，用户仅需要掌握少量的互联网账号，再通过简单的授权操作就能够登录大部分网络应用，这种登录认证方式得到了人们的广泛认同。文献[1]指出，75% 的用户在登录网络应用时会选择 SSO 的方式而不是使用传统的账号口令。但是，服务商在设计开发 OAuth 登录授权 API 时，往往只重视功能性而忽视了安全性。OAuth 身处用户登录授权这一关键节点，一旦出现问题将严重威胁用户隐私，甚至造成大面积的隐私数据泄露。2018 年 1 月，国家信息安全漏洞共享平台（CNVD, China national vulnerability database）发布公告，提醒一个因 OAuth2.0 登录授权 API 实现不当而导致的漏洞 CNVD-2018-01622，该漏洞使攻击者能够在第三方移动应用上非法登录用户账号，从而导致敏感信息泄露。更令人担忧的是，类似的漏洞在 CNVD 上已经多达 44 个。

本文是 OAuth2.0 协议实现安全方面的研究，旨在通过大规模测量揭示因 OAuth2.0 API 脆弱性调用而导致的账号劫持风险，并形成有效的自动化检测工具。本文的主要贡献如下。

1) 分析了 OAuth2.0 协议的风险点，在此基础上提出了基于 OAuth2.0 认证授权的账号劫持威胁模型。

2) 提出了基于差异流量分析的脆弱性 API 识别方法和基于授权认证网络流量检测的账号劫持攻击验证方法，设计实现了面向 OAuth2.0 授权服务 API 的账号劫持威胁检测框架 Oscan。

3) 对 Alexa 排名前 10 000 的网站中真实部署的 3 853 个 OAuth2.0 授权服务 API 进行了大规模检测，发现 360 个 OAuth2.0 API 脆弱性调用。

4) 确认 101 个脆弱性调用可以最终导致账号劫持，涉及 80 个知名网站和 10 个知名 OAuth2.0 服务提供商。

2　相关工作

安全界围绕 OAuth2.0 协议[2]的安全性开展了深入细致的研究，工作主要集中在分析协议的安全性和研究因部署实现而产生的安全问题，后者又可细分为 OAuth2.0 SDK 安全性、OAuth2.0 部署安全性和 OAuth2.0 漏洞自动化检测 3 个方向。各方向的研究工作总结如表 1 所示。

1) OAuth2.0 协议的安全性。此方面的研究方法以模型抽象验证和形式化证明为主要手段，例如 Bansal[3]、Fett[4]、Ferry[5]和文献[6-9]均采用形式化语言描述攻击、建立抽象模型，并以此为基础对大量网站进行安全性验证。此外，也有一些工作尝试从加密证明分析[10]的角度研究 OAuth2.0 协议的安全性。但是上述工作仅在理论上分析 OAuth2.0 协议的安全性，并没有探讨 OAuth 2.0 协议在部署应用过程中可能产生的安全问题。

2) OAuth2.0 SDK 安全性。OAuth2.0 软件开发工具包（SDK, software development kit）是由身份提供方（IdP, identity provider）提供给依赖方（RP, relying party）的开发工具包。若 SDK 存在安全漏洞，则使用该 SDK 进行开发的 RP 都将面临安全风险。此方面的重要研究工作有 Wang 等[11]使用的构建语义模型方法和 Yang 等[12]使用动态符号执行方法实现的工具 SK3Vetter，前者验证了隐含假设的 SDK 存在安全风险，后者则发现了流行 OAuth2.0 SDK 的 7 类逻辑漏洞。OAuth2.0 SDK 安全性研究面临的最大困难是不同厂商的 SDK 差异性非常大，并且往往只面向某种特定的开发语言，因而对于每种 SDK 都需要编写专用的检测工具，并且工具也不方便扩展。

表 1 OAuth 协议已有安全研究工作总结

研究方向	研究进展	研究团队	存在问题
OAuth2.0 协议的安全性	使用形式化描述攻击，提取模型进行检测	BITS Pilani-Goa	未对真实部署的安全性进行研究，仅理论层面进行分析
	建立 Web 模型，形式化分析 OAuth 协议安全性	德国特里尔大学	
	使用加密证明方法，分析 OAuth 协议的安全性	IBM T.J.Watson 研究中心	
OAuth2.0 SDK 安全性	使用语义模型，识别 SDK 中的缺陷	微软雷德蒙德研究院	未覆盖所有的 SDK，工具不易扩展
	利用动态符号执行实现 SK3Vetter，发现逻辑漏洞	香港中文大学	
OAuth 协议部署安全性	检查 state 参数来发现 CSRF 攻击	佐治亚理工学院、武汉大学	手工分析的方法，无法大规模分析
	引入恶意 IdP，发现 4 种攻击方式	波鸿鲁尔大学	
	嗅探 IdP 账号 cookie，进行账号劫持攻击	伊利诺伊大学芝加哥分校	
	发现隐式流程下的 APP 中间人攻击	香港中文大学	
OAuth 漏洞自动化检测	解析视图中 UI 元素，自动化识别访问控制漏洞 AuthScope	德克萨斯大学达拉斯分校	针对于移动应用平台，覆盖的 IdP 较少
	实现了自动化 GUI 测试检测 Facebook API 的工具 SSOScan	弗吉尼亚大学	
	利用静态程序分析与动态行为探测，实现 AuthScan	新加坡国立大学	
	结合协议规范与网络追踪，实现自适应模型框架 OAuthTester	香港中文大学	

3) OAuth2.0 部署安全性。IdP 和 RP 在共同实现 OAuth2.0 协议时，产生了多种安全风险。针对 OAuth2.0 实现不当引发的跨站请求伪造（CSRF, cross-site request forgery）攻击，Shernan 等[13]和 Li 等[14]分析了忽视 state 参数设置的情形下引发的攻击，王丹磊等[15]和 Qiu 等[16]分析了 OAuth2.0 授权 API 中重定向参数的情况。针对 OAuth2.0 账号劫持攻击，Mainka 等[17]利用恶意 IdP 收集用户的 IdP 账号密码从而实施账号劫持攻击，Ghasemisharif 等[18]则通过嗅探用户 IdP 账号的 cookie 重绑定用户账号进行账号劫持攻击。以上 2 种攻击方式都是基于 OAuth2.0 授权码模式。针对 OAuth2.0 隐式授权模式场景下，Hu 等[19]提出了 APP 中间人攻击，Wu 等[20]通过研究 Dropbox 的 SSO 安全风险提出了未授权访问攻击。这些研究通过对真实部署 OAuth2.0 协议的应用程序进行手工分析，发现了多种漏洞。

4) OAuth2.0 漏洞自动化检测。手工分析 OAuth2.0 漏洞显然无法满足大规模的安全检测需求，因而很多工作针对自动化检测 OAuth2.0 漏洞进行了研究。部分工作利用解析视图的 UI 元素实现自动化登录，从而实现自动化地漏洞检测。代表性的工作有 Zuo 等[1]设计的自动化工具 AuthScope 和 Zhou 等[21]实现的自动化检测工具 SSOScan。前者用于识别移动应用程序中访问控制漏洞，后者用于自动化检测 Facebook API 的安全性。另外，部分工作利用网络流量分析实现漏洞的自动化检测，代表性的工作有 Bai 等[22]设计实现的 AuthScan 和 Yang 等[23]提出的 OAuthTester。前者结合动态行为探测与静态程序分析发现了 OAuth2.0 协议部署中的 7 个安全缺陷，后者基于自适应模型对 OAuth2.0 协议进行了 CSRF 风险检测。以上自动化漏洞检测工具对 OAuth2.0 多种安全威胁进行了大量的检测，但只针对一个或少数几个 IdP。

为了弥补现有的 OAuth2.0 漏洞自动化检测的不足，本文提出了针对账号劫持攻击的威胁检测框架 OScan，并在 Alexa 排名前 10 000 的网站上进行测试，覆盖了 26 个 OAuth 授权登录接口服务商。

3 账号劫持风险分析

OAuth2.0 协议描绘了一个完整安全的第三方认证授权的过程，但是协议的实现却可能引入不可预知的安全风险。本节将在回顾 OAuth 认证授权过程的基础上分析 OAuth2.0 协议存在的风险点，并进一步提出基于 OAuth2.0 认证授权的账号劫持威胁模型。该模型首先描述了一种新的账号劫持攻击方式，然后从攻击者角度描述形成此类威胁的必要条件。

3.1 OAuth2.0 协议风险点分析

OAuth2.0 协议提供了 4 种授权模式[2]：授权码模式、简化模式、密码模式和客户端模式。其中授权码模式在互联网上的应用最广泛，该模式下 API 的安全问题也是本文研究的主要内容。图 1 以用户

采用 OAuth 方式登录某个网站为例，简要描述了授权码模式下的授权认证过程。其中，RP 代表用户即将登录的网站，该网站使用了 OAuth 授权登录服务；IdP 代表 OAuth 接口服务提供商，是 OAuth2.0 API 的提供方。

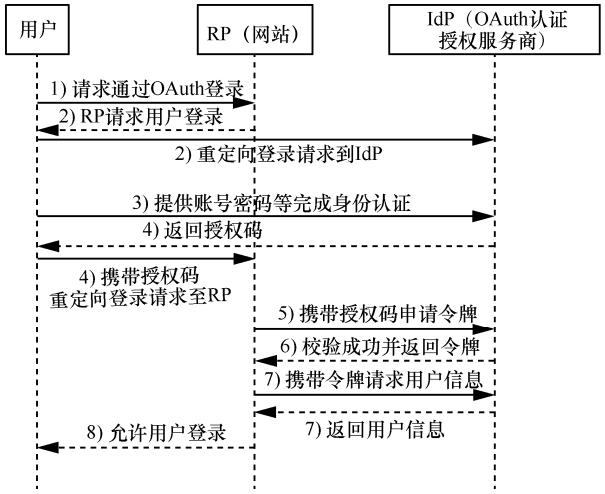


图 1　OAuth2.0 授权码模式认证授权过程

认证授权的具体步骤是：1) 用户访问网站并请求通过 OAuth 登录；2) RP 通过用户重定向 OAuth 登录请求至 IdP，要求身份认证；3) 用户通过账号密码等在 IdP 完成用户身份认证；4) IdP 返回登录授权码，同时重定向登录请求至 RP；5) RP 携带授权码向 IdP 申请访问令牌；6) IdP 校验授权码成功并返回访问令牌；7) RP 持访问令牌从 IdP 获得用户信息；8) RP 允许用户登录。

在整个认证授权过程中，步骤 4)至关重要，处理不当将产生账号被劫持的风险。授权码是用户登录的关键凭据，但是窃取授权码只是劫持账号的充分不必要条件，因为授权码有 2 个特点：1)归属性，即只能被发起认证请求的 RP 使用，在其他 RP 上无效；2)时效性，OAuth2.0 安全规范[24]中指出授权码正确使用后则立即失效，无法二次使用。因此，步骤 4)中 IdP 返回的重定向地址是保证授权码被正确使用的关键，确保合法的 RP 无法获得和使用授权码。只有在这种情况下，恶意第三方劫持到的授权码才是可用的。

大多数网站的开发人员都会严格限制携带授权码的重定向链接，但是目前依然存在一些方法可以有针对性地绕过限制。本文并不讨论如何绕过重定向链接的限制，而侧重于真实互联网上评估已突破这一限制前提下的账号劫持风险。

3.2　基于 OAuth2.0 认证授权的账号劫持威胁模型

根据 3.1 节对于 OAuth2.0 协议风险点的分析，本节提出了一种新的账号劫持攻击方式，并构建了围绕授权码的账号劫持威胁模型。从攻击者角度出发，若想成功劫持 OAuth 登录账号，需要具备 3 个条件。

1) OAuth 认证成功后返回的授权码可窃取。在 OAuth2.0 协议中，授权码是用户通过 IdP 认证并获得登录授权的唯一凭证，授权码对账号劫持的重要性毋庸赘言，关键在于攻击者如何窃取这一认证授权的核心数据。从网络流量的角度看，OAuth2.0 是在 HTTP 基础上的应用实现，具有明文通信这一先天性脆弱点，因此授权码可以被中间人监听；从业务流程角度看，授权码必须先抵达用户浏览器，再以重定向的方式返回给 RP，这又势必面临跨站脚本攻击（XSS, cross site scripting）的威胁。概括起来，授权码被窃取并不是一个独立的安全风险，而是伴生于其他安全风险下的数据泄露问题。

2) OAuth 认证成功后返回的重定向链接可劫持。图 1 步骤 4)中，用户在 IdP 上认证成功后，IdP 为其生成授权码，该授权码连同重定向至网站的链接一同返回给用户。如 3.1 节所述，授权码具有时效性，一旦 RP 使用该授权码向 IdP 申请访问令牌，则授权码立即失效。这一过程是在用户浏览器和后端服务器的配合下自动完成的，如果攻击者不能劫持携带授权码的重定向链接，那么即使窃取到授权码也来不及使用。

授权码的归属性要求其只能被发起请求的 RP 使用，意味着只要攻击者将重定向链接劫持到任意其他非法地址就能保持授权码的可用性。虽然 OAuth2.0 威胁规范中已经充分考虑到这一风险，并在 IdP 和 RP 参数设计规范中禁止跨域的重定向地址，但是出于功能性的考虑，RP 在设计重定向链接参数值时往往不设定唯一的重定向地址，而是给出一个本 RP 的某个目录地址，只检查重定向链接的前缀是否符合要求。这就扩大了合法重定向参数的限定范围，给了攻击者可乘之机。

本文采用的劫持重定向链接的方法正是利用了上述“漏洞”，并不会将重定向地址劫持为一个跨域地址，而是将其劫持为一个既符合 IdP API 规范却又和原 RP 给出的地址不相同的地址，即本域下的某个站点。由于未进行跨域修改，该做法往往

被 RP 和 IdP 认为是合规的。

3) RP 无其他机制二次校验用户登录的合规性。当攻击者窃取到有效的授权码时，仅凭授权码而无其他校验机制成功登录也是一个很大的风险。大量实验表明，一个满足上述 2 个条件的授权码未必能够在攻击者的浏览器上成功登录 RP，例如攻击者 M 窃取了合法用户 A 登录 RP_1 的授权码，虽然该授权码尚未失效，但是 M 在使用该授权码登录 RP_1 时会引发异常而导致登录失败。这一情况又与 IdP 无关，因为同样的场景应用于 RP_2 时，就可能登录成功。因此，本文推测这是由 RP 采用了其他机制二次校验用户登录的合规性，例如 cookie 验证。至于推测是否正确及有哪些二次检验机制，并不是本文所要探讨的内容。本文关注的是直接使用带授权码的登录请求不需要 cookie 验证即可成功登录账号的 RP。

根据上述 3 个条件，本文建立了基于 OAuth2.0 认证授权的账号劫持威胁模型，并以有限状态自动机表示为图 2 的形式。IdP 完成用户的身份认证之后就生成了相应的授权码，如果 RP 或 IdP 上存在修改重定向链接的风险，则该授权码就转变为“对攻击者有效”的状态，否则相继转变为“对攻击者无效”和“用户账号劫持失败”状态；如果在整个授权码传输过程中存在中间人、XSS 等带外数据（OOB, out of band）攻击风险，则该授权码就转变为“攻击者可窃取”的状态，否则相继转变为“攻击者不可窃取”和“用户账号劫持失败”状态；如果 RP 上不存在严格的用户登录合规性校验，则该授权码转变为最终的“可成功劫持用户登录”的状态，否则转变为“用户账号劫持失败”状态。与利用窃取 IdP 账号相关信息（如 cookie 等手段）实现的账号劫持攻击方式相比，该模型描述了一种利用条件相对容易的、新的账号劫持的攻击方式。

3.3 账号劫持威胁检测方法

针对 3.2 节提出的劫持 OAuth 登录账号的必要条件，本节将提出相应的检测方法。在判断 OAuth 认证成功后返回授权码是否可劫持账号方面，本文借鉴传统的 Web 安全检测方法，提出了基于差异流量分析的脆弱性 API 识别方法和基于授权认证网络流量监测的账号劫持攻击验证方法。

1)基于差异流量分析的脆弱性 API 识别方法。对于同一个 RP 的相同 OAuth API 调用，构造异常重定向地址参数，并分别自动化触发正常调用和多个异常调用，通过比较返回值的差异，判断认证成功后返回的重定向链接是否可劫持，从而确定该 API 调用是否具有脆弱性。

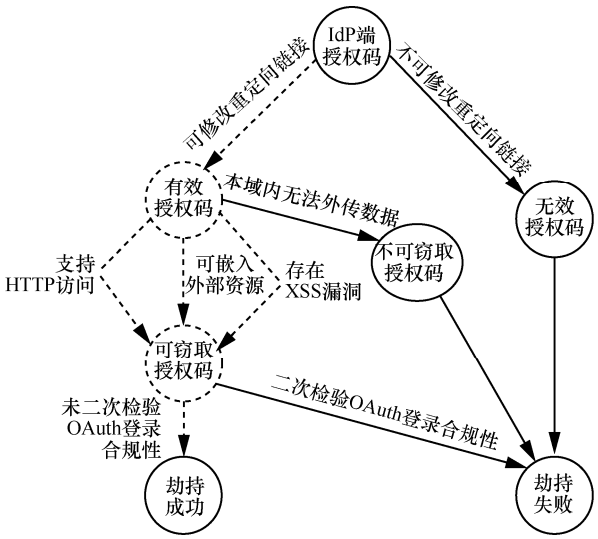


图 2 基于 OAuth2.0 认证授权的账号劫持威胁模型

2) 基于授权认证网络流量检测的账号劫持攻击验证方法。通过拦截和重放用户真实的 OAuth 登录请求，实际验证使用劫持的授权码登录账号的过程，确认 RP 是否存在用户登录的合规性校验机制。

4 OScan 设计与实现

围绕基于 OAuth 授权码的账号劫持攻击威胁模型及检测方案，本文设计实现了面向 OAuth2.0 授权服务 API 的账号劫持威胁检测框架 OScan。OScan 由 OAuth API 检测、OAuth2.0 API 脆弱性调用检测、OOB 攻击风险检测和授权码登录 4 个子系统构成。除 OAuth API 检测子系统外的 3 个子系统，分别检测目标网站是否满足 3.2 节提出的账号劫持必要条件： OAuth 认证成功后返回的授权码可窃取、OAuth 认证成功后返回的重定向链接可劫持和 RP 无其他机制二次校验用户登录的合规性。OScan 的系统架构如图 3 所示。

4.1 OAuth API 检测子系统

OAuth API 检测子系统由网页爬虫、流量捕获器、分析引擎和 OAuth API 模式库 4 个模块构成，提供检测某个网站是否支持 OAuth 登录认证，判断提供 OAuth 服务的 IdP 和提取 OAuth API 这 3 个方面的功能。

网页爬虫以目标网站的首页作为输入，功能

是触发页面中全部的点击事件，产生网络流量。文献[10]通过静态匹配当前页面 URL 中的关键字（如 client_id、grant_type）来判断是否发生 OAuth API 调用，但是一些 RP 在调用 OAuth API 时会产生页面 URL 的跳转，因此这种方法会产生很高的漏报率。与文献[25]提出的基于规则库的网络爬虫方法相似，OScan 采用更加精准的捕获并分析实际流量的方法检测网页中的 OAuth API 调用。流量捕获器实际上是一个 HTTP 和 HTTPS 代理，可以捕捉到网页爬虫触发的全部流量，并输入给分析引擎。分析引擎提取其中的请求地址，并在 OAuth API 模式库中查找匹配，检测是否触发了 OAuth API 调用。OAuth API 模式库收集了来自 26 个知名 IdP 的全部 OAuth API 的模式特征。

这 26 个 IdP 由维基百科公布的国际知名 OAuth IdP 和国内知名 OAuth IdP 共同组成，覆盖面广泛。如果当前页面中未能发现 OAuth API 调用，网页爬虫还将按照预置列表搜索登录关键字（如 login、signin、oauth、log-in、sign-in 等），OScan 还将尝试在首页的下一级发现登录页面。算法 1 给出了 API 提取的伪代码描述。

算法 1　API 提取

输入　目标网站，API 模式库，登录关键字

输出　OAuth 授权 API

```
1) 渲染目标网站首页，构建 DOM 树
2) for DOM 节点 in DOM 树 do
3)     触发 DOM 节点的点击事件
4)     for 匹配规则 in API 模式库 do
5)         if 流量 match 匹配规则 then
6)             输出该 API 调用
7)         end if
8)     end for
9)     for 关键字 in 登录关键字表 do
10)        if 关键字 in 流量 then
11)            将其作为新目标重复步骤 1)~步骤 14)
12)        end if
13)    end for
14) end for
```

4.2　OAuth2.0 API 脆弱性调用检测子系统

OAuth2.0 API 脆弱性调用检测子系统的任务是检测特定的 OAuth2.0 API 是否存在可劫持重定向参数的漏洞，由参数定位、参数修改和差异分析 3 个功能模块构成。

OScan 检测 OAuth2.0 API 调用脆弱性的思路如下。首先，参数定位模块以实际 OAuth2.0 API 调用流量为输入，识别其中的各个参数，尤其是精准定位重定向地址参数。由于 OScan 只面向 26 个 IdP 提供的有限 API，因此对各个参数的定位采用了枚举的方法。其次，参数修改模块将重定向参数修改为同一父域下的不同子域或同一域下的不同页面，并重新向 IdP 发起请求。为覆盖更多的测试路径，该模块借鉴“鸟枪法”思路，将多次尝试，分别将重定向参数修改为不同的值。最后，模块接收正常传参和异常传参下 API 调用的响应返回值，本文利用差分网络流量分析法进行脆弱性识别。即如果检测到任意一次异常传参后的 API 返回内容与正常传参的情况相同，则表明检测到了 OAuth2.0 API 脆弱性调用，否则不存在脆弱性调用。算法 2 给出了 OAuth2.0 API 脆弱性调用检测的伪代码。

算法 2　OAuth2.0 API 脆弱性调用检测

输入　OAuth2.0 API 调用列表

输出　脆弱的 OAuth2.0 API 调用

```
1) for API 调用 in OAuth2.0 API 调用列表:
2)     回调值=API 调用中’redirect_uri’参数值
3)     前，中，尾=以回调值划分的 API 调用
4)     域名=回调值中的域名
5)     链接=爬取域名网站
6)     for 链接 1in 链接:
7)         if 链接 1 != 回调值:
8)             新请求=前+链接 1 +尾
9)             响应 1=发送新请求
10)            响应 2=发送原来的请求
11)            if 响应 1==响应 2:
12)                该 OAuth2.0 API 调用具有脆弱性
13)                break
14)            end if
15)        end if
16)    end for
17) end for
```

4.3　OOB 攻击风险检测子系统

OOB 攻击并没有公认的准确定义。本文使用这一术语表示“通过信息系统预定功能之外的方式非法传输数据的技术或行为”，例如 Web 安全领域的

SQL 注入、XSS 和 XXE 可都能引发 OOB 风险，文献[26]也提到了利用 XSS 的方式。

OScan 中实现的 OOB 攻击风险检测子系统可以自动检测 3 项风险：网站是否存在 XSS 漏洞、网站是否可嵌入外部资源和网站是否可以使用 HTTP 访问。检测到上述任意一种风险，都表明攻击者有机会窃取到 OAuth2.0 API 返回的授权码。这部分工作在 Web 安全领域已有成熟的技术[27]和方法，本文不再详述。

4.4 授权码登录子系统

OScan 的最后一步工作是尝试使用“劫持”的授权码自动登录网站，并通过与此前人工登录成功的页面对比，判断劫持账号是否成功。如果成功，则确认网站存在基于 OAuth2.0 认证授权的账号劫持风险。

本文在授权码登录子系统中利用流量分析法来验证使用授权码登录的安全威胁，由授权登录和验证 2 个模块组成。由于涉及自动化登录，授权模块针对 26 个 IdP，首先收集了其账号的 cookie 信息，使其不需要输入账号密码即可成功完成认证。授权登录模块使用 Burp Suite 实现。它模拟用户发送第三方登录认证请求（携带账号 cookie 信息），自动进行登录授权。然后从拦截的流量中通过识别“code”关键字提取出带有授权码的登录请求，将其发送至验证模块，验证模块收到来自授权模块的请求，向 RP 发送登录请求（不带 cookie 信息），获取流量。

若流量中存在类似“remember_user_token”令牌，如图 4 所示，则验证了该 RP 存在账号劫持风险；反之，则说明不具有该风险。算法 3 给出了账号劫持攻击验证算法的伪代码。

```
GET / HTTP/1.1
Host:www.xxxxx.com
Connection: close

Cookie: Signin_redirect= https%3A%2F%2Fwww.xxx.com%2F;
Hm_lpvt_0c0e9d9b1e7d617b3e6842e85b9fb068=1552742341;remember_user_token=W1sxNDxxxxxxxxxxYSQxMSRSekp3MWFORy9MMEl1QS9lNjJnb3JPIiwixxxxxxxxxxxjc5OC4xMDAxOTA5Il0%3D--xxxxxxxxxxxx5defac19492ec3bcbd00fbe6a3ad
```

图 4 验证成功登录流量

算法 3 账号劫持攻击验证

输入 RP 列表，IdP 账号 cookies

输出 存在账号劫持风险的 RP

1) for RP in RP 列表:
2) for cookie in IdP 账号 cookies:
3) 向 RP 发送带有 cookie 的登录请求
4) if code 参数 in 流量:
5) 请求 1=带 code 的认证请求
6) end if
7) 向 RP 发送请求 1，获取响应
8) if remember_user_token in 响应:
9) RP 存在账号劫持风险
10) else
11) RP 不存在账号劫持风险
12) end if
13) end for
14) end for

5 效果评估

本节基于第 4 节设计的 OScan 系统，选取 Alexa 排名前 10 000 的网站作为测试对象，从是否存在登录劫持风险的角度评测互联网上 OAuth 认证授权的安全现状。

5.1 OAuth 应用现状

对 Alexa 排名前 10 000 的网站进行测量，结果如图 5 所示，得到 OAuth 应用现状如下。

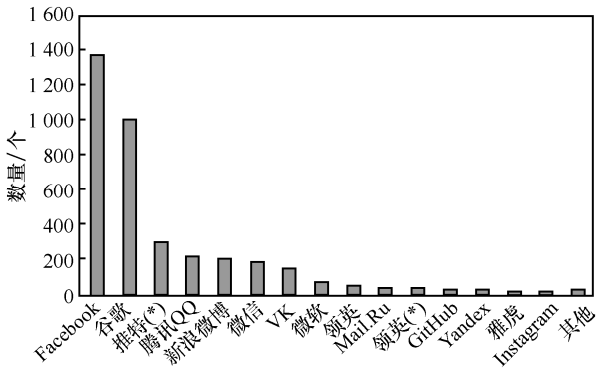


图 5 API 调用流行度

1) Alexa 排名前 10 000 的网站中，有 1 881 个网站支持 OAuth 的认证登录，占比接近 20%，可见 OAuth 确实是颇受欢迎的 SSO 解决方案。

2) 在 1 881 个支持 OAuth 认证登录的网站中，累计检测到 3 853 个 OAuth API 调用，这表明平均每个网站提供 2 种不同的 OAuth 登录认证服务供用户选择。

3) Facebook 和谷歌是最受青睐的 IdP，市场占比分别达到 35.61%和 26.27%，并且参与测试部国外网站几乎都支持以 Facebook 和谷歌账号登录。

4) 包括推特、领英在内的若干个 IdP 仍然提供基于 OAuth 1.0a 协议实现的 API（图 5 中带*的 IdP），

并且 3 853 频次的 OAuth API 调用中也有 402 频次采用了 OAuth 1.0a 协议，可见老旧的 OAuth 1.0a 协议仍然占据一定的市场份额。

虽然测量结果表明仍有相当部分网站使用 OAuth 1.0a 协议认证授权，但更新、更安全的 OAuth2.0 协议取代 OAuth 1.0a 协议已经是大势所趋，因此本文后续工作只针对其余 3 451 个 OAuth2.0 API 调用做进一步的登录劫持风险分析。

5.2 脆弱性 API 调用分析

本文在 3.2 节提出了劫持 OAuth 登录账号的 3 个必要条件，其中，只有条件 2)与 OAuth API 相关，造成重定向链接被劫持的原因既可能是 IdP 检查重定向参数不严格，也可能是 RP 未能严格遵循 API 调用规范，本文将这 2 种情况统称为“脆弱性 API 调用”，可以视为一种漏洞。

实验结果如表 2 所示。在 OScan 检测的 3 451 个 OAuth2.0 API 调用中，有 360 个是脆弱性调用，占总数的 10.43%，并且个别 IdP 提供的 OAuth2.0 API 脆弱性调用竟然超过 50%。由此可见，脆弱性 OAuth2.0 API 调用不仅普遍存在，而且在个别 OAuth 服务中还相当严重。从测量结果看，并不是隶属于同一 IdP 下的全部 API 调用都存在问题，这说明脆弱性 API 调用并非 IdP 或 RP 单方原因，而是双方共同导致的。因此，IdP 开发 OAuth2.0 API 和 RP 调用 OAuth2.0 API 时，都应当在满足功能性需求的同时充分考虑安全性需求。

5.3 登录劫持风险分析

本节尝试检测满足劫持 OAuth 登录账号全部 3 个必要条件的网站，显然是 5.2 节分析结果的子集。为满足劫持 OAuth 登录账号的条件 1)，OScan 检测至少符合以下一种情况的网站：网站存在可嵌入外部资源的漏洞以及 IdP 与 RP 间的通信采用 HTTP，是明文流量。为满足劫持 OAuth 登录账号的条件 3)，本文在每个 IdP 上都注册了大量账号，OScan 将使用劫持的授权码模仿攻击者尝试实际登录。

实验结果如表 3 所示。全部 360 个存在脆弱性 OAuth2.0 API 调用中，有 101 个调用可以被攻击者成功用于劫持登录账号，占比达到 28.05%。这 101 个 API 调用覆盖了 80 个网站，涉及 10 个著名的 IdP。虽然存在 OAuth 登录劫持脆弱性的网站比例不高，但是泄露的隐私可能进一步导致用户 IdP 账号被窃取，由此引发的次生安全问题将不可估量。

表 2　脆弱性 API 调用结果

IdP	API 调用/个	脆弱性调用/个	占比
Facebook	1 372	132	9.62%
谷歌	1 012	0	0
推特(*)	306	0	0
腾讯 QQ	224	6	2.68%
新浪微博	208	117	56.25%
微信	183	27	14.75%
VK	159	46	28.93%
微软	69	4	5.80%
领英	56	0	0
Mail.Ru	44	19	43.18%
领英(*)	44	0	0
GitHub	32	4	12.50%
Yandex	28	0	0
雅虎	22	3	13.64%
Instagram	13	0	0
亚马逊	10	0	0
Twitch	7	0	0
百度	4	2	50.00%
Foursquare	3	0	0
Tumblr(*)	2	0	0
Battle.net	2	0	0
Autodesk(*)	1	0	0
Dropbox	1	0	0
ALO	1	0	0
Reddit	1	0	0
印象笔记(*)	1	0	0

表 3　API 调用登录劫持风险结果

IdP	脆弱性调用/个	攻击成功数量/个	占比
Facebook	132	20	15.15%
新浪微博	117	46	39.32%
VK	46	16	34.78%
微信	27	17	62.96%
Mail.Ru	19	0	0
腾讯 QQ	6	1	16.67%
微软	4	1	25.00%
Github	4	0	0
雅虎	3	0	0
百度	2	0	0
总和	360	101	28.05%

表 4 自动化检测工具对比

工具	输入	IdP 数量/个	RP 数量/个	可检测漏洞类型	脆弱点数量/个	完整攻击风险数量/个	可否扩展
SK3Vetter[12]	SDK	10	无	逻辑错误	7	—	否
AuthScope[1]	Android APP	1	4 838	信息泄露	306	—	否
SSOScan[21]	网站、网络流量	1	10 000	信息泄露	202	—	否
OAuthTester[12]	OAuth2.0 协议、网络流量	4	500	CSRF 攻击	223	—	是
OScan	目标网站首页	26	10 000	账号劫持攻击	360	80	是

5.4 OScan 的优势分析

本节将 OScan 与其他 OAuth 漏洞自动化检测工具进行对比，结果如表 4 所示。通过对输入、IdP 数量、RP 数量、可检测漏洞类型、脆弱点数量、完整攻击风险数量、可否拓展 7 个方面进行比较，发现 OScan 在以下 4 个方面都具有优势。

1) OScan 在覆盖的 IdP 上具有全面性。相较于其他工具仅考虑了一个或少数几个 IdP，OScan 覆盖了 26 个 IdP，包括主流的 Facebook、谷歌等。

2) OScan 对广泛的 RP 进行了检测。OScan 对 Alexa 排名前 10 000 的网站进行了检测，大部分的工具所检测的 RP 数量远远少于 OScan 的这一数量。

3) OScan 能够检测 OAuth2.0 中新的安全风险点。OScan 能识别出 360 个具有账号劫持攻击风险的脆弱点，这种能力是其他工具不具备的。

4) OScan 能够检测完整的攻击风险。OScan 在识别出脆弱性 API 的基础上，还对由此引发的账号劫持攻击进行了检测，这也是其他工具所不具备的能力。

综上，与已公开的类似工具相比，OScan 在覆盖的 IdP 数量、检测的 RP 数量、检测完整攻击风险等方面均具有非常明显的优势。

5.5 案例分析

某网站支持某 IdP OAuth 认证授权，其 API 调用如图 6 所示。其中，client_id 参数表示客户端的 ID，用于标识网站应用；redirect_uri 参数表示重定向链接；response_type 参数表示授权类型；此处的值“code”表示采用授权码模式；state 参数表示客户端的当前状态值，可以设定为任意值，认证服务器会原封不动地返回该值，其作用是防止 CSRF 攻击。

```
https://api.xxx.com/oauth2/authorize?client_id=xxxx
xxx&redirect_uri=http://www.xxxxx.com/users/auth/w
eibo/callback&response_type=code&state=aaxxxxces
```

图 6 某网站调用某 IdP OAuth 认证授权 API 实例

OScan 分析表明，该 API 中的重定向地址（redirect_uri 参数）可以被替换为本站的其他地址，且认证过程采用了未加密的 HTTP。图 7 是重定向地址被劫持后的 API 调用。

```
https://api.xxx.com/authorize?client_id=xxxx
xxx&redirect_uri=http://www.xxxxx.com/p/1fc5dfb4
4e8f&response_trpe=code&state=aaxxxxecs
```

图 7 攻击者利用 XSS 漏洞读取授权码

OScan 分析还表明，劫持后的重定向地址 http://www.xxxxx.com/p/1fc5dfb44e8f 恰好存在 XSS 漏洞。因此，攻击者可以通过钓鱼邮件诱使用户点击链接发起 OAuth 认证，从而窃取 IdP 返回的授权码，完成登录劫持。图 8 展示了攻击者利用 XSS 漏洞读取的授权码。

```
https://xxxxx.com/user/oauth/xxx?code=f300686c
dxxxxxxxxxx4b7dcc9c811
```

图 8 授权码登录接口

6 讨论

6.1 局限性分析

本文研究探讨了 OAuth2.0 API 的安全问题，构建了基于授权码的账号劫持攻击模型，并在此基础上设计实现了自动化检测工具 OScan。虽然本文利用 OScan 对 10 000 个网站进行了大规模的账号劫持风险检测，并得到了实验结论，但本文的工作仍然具有一定的局限性。首先，仅针对基于授权码模式的 OAuth2.0 授权 API 进行安全性分析与检测，没有对其他授权模式（如隐式模式）下的授权 API 进行研究；其次，未覆盖其他平台（如 Android、iOS 等）应用程序的授权 API；最后，OScan 在技术细节上仍有提高空间，例如 API 提取的结果会因网络状况受到影响，且在提取出网站登录链接时所匹配的关键字单词集可能不够完善，因此会存在一定的 API 遗漏的情况。

6.2 缓解措施

为防御针对 OAuth2.0 授权 API 的账号劫持攻

击，第三方应用程序在实现授权码模式过程中必须严格地遵循规范，本文建议应做到以下 3 点。1)严格限制重定向链接参数的值。为了防止攻击者篡改重定向链接的值，RP 不应当将重定向的值注册为某个目录，而是指定唯一地址。对于 IdP 而言，应当在开发文档中要求开发人员将重定向链接的参数设置为唯一确定值。2)提高 RP 和 IdP 自身的安全性。包括严格限制外部资源的嵌入、全面排查和修复 XSS 漏洞，防止攻击者通过 OOB 的方式窃取授权码；整个授权认证过程都部署 HTTPS，以通信内容加密的方式确保授权码无法被攻击者窃取。3)对 OAuth2.0 授权认证过程进行完整性校验，如 RP 主动检查用户 cookie。在 RP 向 IdP 申请认证授权时，RP 需建立申请认证的用户与某个指定参数之间的联系（如 state 参数），在后续 RP 接收到带有授权码的授权请求时，将用户与提交的某个指定参数进行校验，只有符合关联的请求方可继续向 IdP 申请令牌，防止攻击者使用窃取的授权码直接成功登录用户账号。

7　结束语

在第三方登录变得越来越受欢迎的同时，安全风险也随之而来。基于此，本文对 OAuth 2.0 授权 API 的安全性进行了深入研究，构建了基于授权码的账号劫持攻击模型。此外，为了检测这一安全威胁，本文设计实现了针对 OAuth2.0 授权服务 API 的威胁检测框架 OScan，该工具通过差异流量分析方法识别出脆弱性 API，通过基于授权认证网络流量监测的方法进行账号劫持攻击验证。为了评估互联网上 OAuth2.0 授权 API 的安全现状，本文对 Alexa 排名前 10 000 的网站 OAuth2.0 授权 API 进行了实验。结果表明，10.43%的 API 存在重定向链接可被修改的脆弱点，其中 80 个第三方网站的用户账号都可以被成功劫持，进而获取用户隐私信息，存在着严重的安全隐患。开发者和 IdP 都应对此引起重视，严格遵守协议规范，对各个脆弱点进行防范，保证其安全性。

本文所实现的 OScan 虽然对 Alexa 排名前 10 000 的网站中 OAuth2.0 授权服务 API 的账号劫持攻击威胁进行了安全性检测测量，但未来仍有提升空间，包括实现对 OAuth2.0 协议的其他授权模式（如隐式模式）、其他平台（如移动应用）的安全性分析，使 OScan 成为适应多平台、多授权模式的 OAuth2.0 账户劫持攻击威胁检测工具。

参考文献：

[1] ZUO C, ZHAO Q, LIN Z. Authscope: towards automatic discovery of vulnerable authorizations in online services[C]//The 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 53-68.

[2] HARDT D. The OAuth 2.0 authorization framework[Z]. RFC6749, 2012.

[3] BANSAL C, BHARGAVAN K, DELIGNAT-LAVAUD A, et al. Discovering concrete attacks on website authorization by formal analysis[J]. Journal of Computer Security, 2014, 22(4): 601-657.

[4] FETT D, KUSTERS R, SCHMITZ G. A comprehensive formal security analysis of OAuth 2.0[C]//The 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 1204-1215.

[5] FERRY E, O RAW J, CURRAN K. Security evaluation of the OAuth 2.0 framework[J]. Information & Computer Security, 2015, 23(1): 73-101.

[6] 魏成坤, 刘向东, 石兆军. 基于 OAuth2.0 的认证授权技术研究[J]. 信息网络安全, 2016(9): 6-11.
WEI C K, LIU X D, SHI Z J. Optimization method for OAuth2.0 protocol[J]. Netinfo Security, 2016(9): 6-11.

[7] 魏成坤, 刘向东, 石兆军. 基于 OAuth2.0 协议的安全性形式化分析[J]. 计算机工程与设计, 2016, 37(7): 1746-1751.
WEI C K, LIU X D, SHI Z J. Security formal verification of OAuth2.0 protocol[J]. Computer Engineering and Design, 2016, 37(7): 1746-1751.

[8] 王焕孝, 顾纯祥, 郑永辉. 开放授权协议 OAuth2.0 的安全性形式化分析[J]. 信息工程大学学报, 2014, 15(2): 141-147.
WANG H X, GU C X, ZHENG Y H. Formal security analysis of OAuth2.0 authorization protocol[J]. Journal of Information Engineering University, 2014, 15(2): 141-147.

[9] 郭丞乾, 蔡权伟, 林璟锵, 等. 单点登录协议实现的安全分析[J]. 信息安全研究, 2019, 5(1): 59-67.
GUO C Q, CAI Q W, LIN J J, et al. Security analysis on the implementations of single-sign-on protocols[J]. Journal of Information Security Research, 2019, 5(1):59-67.

[10] CHARI S, JUTLA C S, ROY A. Universally composable security analysis of OAuth v2.0[J]. IACR Cryptology ePrint Archive, 2011: 526.

[11] WANG R, ZHOU Y, CHEN S, et al. Explicating SDKs: uncovering assumptions underlying secure authentication and authorization[C]//The 22nd USENIX Conference on Security. USENIX Association, 2013: 399-314.

[12] YANG R, LAU W C, CHEN J, et al. Vetting single sign-on implementations via symbolic reasoning[C]//The 27th USENIX Security Symposium (USENIX Security 18). USENIX, 2018: 1459-1474.

[13] SHERNAN E, CARTER H, TIAN D, et al. More guidelines than rules: CSRF vulnerabilities from noncompliant OAuth 2.0 implementations[C]// International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. 2015: 239-260.

[14] LI W, MITCHELL C J. Security issues in OAuth 2.0 SSO implementations[C]//International Conference on Information Security. 2014:

529-541.

[15] 王丹磊, 李长军, 赵磊, 等. OAuth2.0 协议在 Web 部署中的安全性分析与威胁防范[J]. 武汉大学学报(理学版), 2016, 62(5): 411-417.
WANG D L, LI C J, ZHAO L, et al. Security analysis and vulnerability management of OAuth 2.0 on Web deployment[J]. Journal of Whhan University (Natural Science Edition), 2016, 62(5): 411-417.

[16] QIU K, LIU Q, LIU J, et al. An empirical study of OAuth-based SSO system on Web[C]//International Conference on Wireless Algorithms, Systems, and Applications. 2018: 400-411.

[17] MAINKA C, MLADENOV V, SCHWENK J. Do not trust me: using malicious IdPs for analyzing and attacking single sign-on[C]//2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016: 321-336.

[18] GHASEMISHARIF M, RAMESH A, CHECKOWAY S, et al. O single sign-off, where art thou? An empirical analysis of single sign-on account hijacking and session management on the web[C]//The 27th USENIX Security Symposium (USENIX Security 18). USENIX, 2018: 1475-1492.

[19] HU P, YANG R, LI Y, et al. Application impersonation: problems of OAuth and API design in online social networks[C]//The Second ACM Conference on Online Social Networks. ACM, 2014: 271-278.

[20] WU B, NGUYEN T, HUSAIN M. Implementation vulnerability associated with OAuth 2.0—a case study on Dropbox[C]//The 12th International Conference on Information Technology-New Generations. 2015: 135-138.

[21] ZHOU Y, EVANS D. SSOScan: automated testing of web applications for single sign-on vulnerabilities[C]//The 23rd USENIX Security Symposium (USENIX Security 14). USE NIX, 2014: 495-510.

[22] BAI G, LEI J, MENG G, et al. AUTHSCAN: automatic extraction of web authentication protocols from implementations[C]//NDSS. 2013.

[23] YANG R, LI G, LAU W C, et al. Model-based security testing: an empirical study on OAuth 2.0 implementations[C]//The 11th ACM on Asia Conference on Computer and Communications Security. ACM 2016: 651-662.

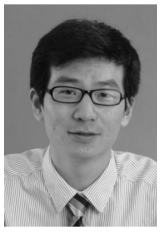
[24] LODDERSTEDT T, MCGLOIN M, HUNT P. OAuth 2.0 threat model and security considerations[Z]. RFC 6819, 2013.

[25] 杜雷, 辛阳. 基于规则库和网络爬虫的漏洞检测技术研究与实现[J]. 信息网络安全, 2014(10): 38-43.
DU L, XIN Y. Research and implementation of web vulnerability detection technology based on rule base and web crawler[J]. Netinfo Security, 2014(10): 38-43.

[26] 陈君, 张生. 基于 OAuth 单点登录系统的安全性分析与评估[J]. 电子科技, 2017, 30(9): 165-168.
CHEN J, ZHANG S. Security evaluations and countermeasures of single sign-on systems based on OAuth protocol[J]. Electronic Science and Technology, 2017, 30(9): 165-168.

[27] 张天琪. OAuth 协议安全性研究[J]. 信息网络安全, 2013(3): 68-70.
ZHANG T Q. Study on OAuth protocol security[J]. Netinfo Security, 2013(3): 68-70.

[作者简介]



刘奇旭（1984− ），男，江苏徐州人，博士，中国科学院信息工程研究所副研究员，中国科学院大学副教授，主要研究方向为网络攻防技术、网络安全评测。

邱凯丽（1996− ），女，土家族，湖南张家界人，中国科学院大学硕士生，主要研究方向为网络攻防技术。

王乙文（1996− ），男，浙江湖州人，中国科学院大学硕士生，主要研究方向为网络攻防技术。

陈艳辉（1996− ），男，山东潍坊人，中国科学院大学博士生，主要研究方向为网络攻防技术。

陈浪平（1995− ），男，浙江绍兴人，中国科学院大学硕士生，主要研究方向为网络攻防技术。

刘潮歌（1986− ），男，吉林长春人，博士，中国科学院信息工程研究所助理研究员，中国科学院大学讲师，主要研究方向为网络攻击追踪溯源、Web 安全和网络欺骗。