单点登录协议实现的安全分析

郭丞乾[1,2,3]　蔡权伟[2,3]　林璟锵[1,2,3]　刘丽敏[2,3]
[1](中国科学院大学网络空间安全学院　北京　100049)
[2](中国科学院数据与通信保护研究教育中心　北京　100093)
[3](信息安全国家重点实验室(中国科学院信息工程研究所)　北京　100093)
(guocqian@gmail.com)

Security Analysis on the Implementations of Single-Sign-On Protocols

Guo Chengqian[1,2,3], Cai Quanwei[2,3], Lin Jingqiang[1,2,3], and Liu Limin[2,3]
[1](*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049)
[2](*Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing* 100093)
[3](*State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences), Beijing* 100093)

Abstract　Web applications rely on the authentication to ensure the security of systems and protect the users' privacy. The single-sign-on (SSO) services, provided by the identity service providers (IdP), allow the Web applicatioins to integrate the authentication directly, instead of maintaining and protecting the users' credentials by themselves. Moreover, SSO systems make it easier for users to visit multiple applications, for example, each user only needs to maintain one credential, and completes the authentication at the chosen IdP. However, various vulnerabilities have been found in the implementations of SSO systems. In this paper, we analyze three mainstream SSO protocols, namely, OAuth 2.0, OpenID-Connect and SAML, and provide the common process for SSO systems. Based on the goals of adversaries and the ability of each participant, we propose four attack scenarios, and present seven security assumptions that should be satisfied in SSO systems. The analysis on existing attacks demonstrate that at least one assumption has been broken for each vulnerability. Our work help to design, implement and analyze secure SSO services.

Key words　authentication; single-sign-on; OAuth 2.0; OpenID-Connect; SAML; security rule

摘　要　身份鉴别是保证信息系统安全性和用户隐私的必要前提,单点登录协议和系统使得专业的身份服务提供商能够在确保用户隐私得到有效保护的前提下,为用户提供良好的体验(无需记忆多个口令,1次登录即可使用多个应用等),同时又避免了网络服务提供商重复开发用户身份鉴别功能.然而,已有研究成果表明单点登录协议在实现时存在诸多安全问题,本次安全分析针对当前主流的单点登录协议,如OAuth 2.0,OpenID-Connect以及SAML,抽象出单点登录协议的基本流程和

收稿日期:2018-11-15
基金项目:国家重点研发计划项目(2017YFB0802100)
通信作者:林璟锵(linjq@is.ac.cn)

攻击者目标;针对协议参与方的能力形成不同的攻击场景,在此基础上形成了单点登录系统必须满足的7条安全假设.针对现有单点登录系统漏洞的分析,表明实际系统中的漏洞均是由于违背了7条安全假设中的1条或多条.对单点登录系统面临的安全问题进行了系统研究,为单点登录系统的设计、实现和分析奠定了基础.

关键词 身份鉴别;单点登录;OAuth 2.0;OpenID-Connect;SAML;安全假设

中图法分类号 TP309.2

随着互联网+时代的到来,网络应用和服务的数量和类型大规模增长.为了保护用户隐私,提供定制化服务,网络应用需要实现用户身份鉴别功能.最初,网络应用系统各自实现用户身份鉴别功能,这种方式一方面增加了网络应用系统的开发和维护工作量;另一方面,提高了用户维护账号信息的难度;同时,由于不同网络应用系统的能力存在差异,其用户隐私信息保护能力不尽相同,而用户为了便利通常可能在不同的网络应用系统中使用了相同、相似的口令,增加了用户隐私泄露的风险.

为保护用户隐私,降低网络应用系统的开发维护难度,多个身份服务提供商,如Google、Facebook、微信、微博等已经提供单点登录功能,即用户仅需在身份服务提供商登录1次,即可使用多个不同的网络应用.单点登录功能使得网络应用系统不再需要维护复杂的身份鉴别子系统;同时使得用户只需记住1套账号,即可使用多个网络应用.身份服务提供商一般为技术能力较强的大型企业,由身份服务提供方负责用户身份的鉴别,能够为用户的隐私信息提供更完善的保护.

单点登录功能涉及到三方,即:身份服务提供商(后文简称为服务提供方)、第三方网络应用(后文简称为依赖方)和用户.本文首先分析了主流的单点登录协议,然后根据攻击者目标以及单点登录协议的基本流程,分析不同应用场景及攻击者能力,提出了单点登录系统需要满足的安全假设,并通过对已有攻击的分析分类,验证了安全假设的合理性,为后续单点登录系统的设计、研究奠定了基础.

1 单点登录协议介绍

OAuth 2.0协议、OpenID-Connect协议、SAML(security assertion markup language)协议是使用最为广泛的单点登录协议.本节将对这3种协议进行介绍.

1.1 OAuth 2.0协议

OAuth 2.0[1]协议最初的设计目标是用户授权,主要功能是允许依赖方向用户资源持有方申请获得用户资源,这一功能通过授予依赖方访问令牌来实现,依赖方可以使用访问令牌向用户资源持有方申请获取用户资源.使用OAuth 2.0协议进行用户身份鉴别时,接入用户身份鉴别服务的依赖方通过访问令牌来获得资源,即用户身份信息.

OAuth 2.0协议最常用的2种模式为隐式模式和授权码模式,具体协议如图1所示,其中虚线代表浏览器的重定向功能,实线代表直接的网络通信,线上的内容代表通信过程中传递的参数.

OAuth 2.0协议隐式模式的流程如下:

① 用户发起向依赖方的访问;

② 依赖方检查用户处于未登录状态,将用户重定向至服务提供方;

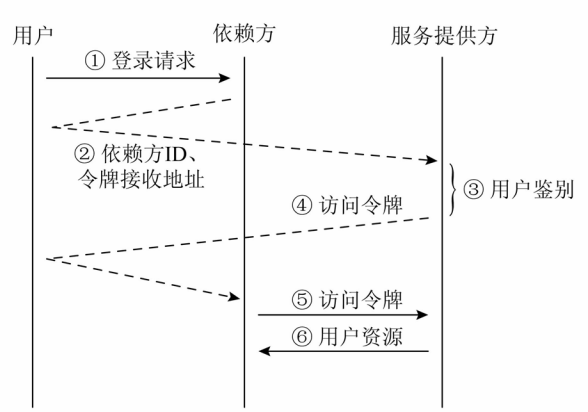
③ 用户在服务提供方完成身份鉴别;

④ 服务提供方将访问令牌重定向至依赖方;

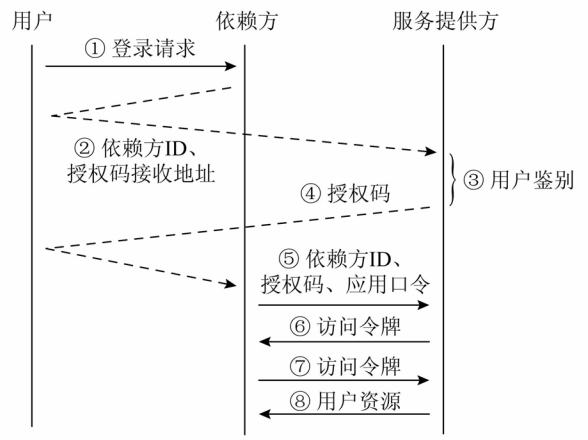
⑤ 依赖方使用访问令牌向服务提供方申请用户身份信息;

⑥ 服务提供方向依赖方发送用户身份信息.

OAuth 2.0协议授权码模式与隐式模式的区别在于授权码模式下,服务提供方并不会直接传递给身份依赖方访问令牌,而是发送一个授权码,之后由依赖方使用授权码以及与服务提供方共享的一个应用口令申请访问令牌(如图1(b)中④~⑥所示).隐式模式与授权码模式在设计上的不同导致安全问题,我们会在后文的安全分析中详细说明.



(a) 隐式模式



(b) 授权码模式

图 1　OAuth 2.0 协议流程

1.2　OpenID-Connect 协议

由于 OAuth 2.0 协议本身是作为授权协议而产生的，其作为身份鉴别协议存在缺陷[2]，详细问题在后文会进行讨论. 为此，OpenID-Connect 作为 OAuth 2.0 协议的扩展协议[3]被提出，以专门用于身份鉴别.

OpenID-Connect 协议与 OAuth 2.0 协议流程基本相同，主要区别是身份鉴别请求中添加 openid 参数，服务提供方最终会发送给依赖方一个 JWT[4]格式的身份令牌，依赖方可以通过身份令牌鉴别用户身份，流程如图 2 所示.

OpenID-Connect 协议中的身份令牌与 OAuth 2.0 的访问令牌的主要区别在于：访问令牌只是一个随机字符串，而身份令牌则携带了依赖方、服务提供方、用户等信息并进行了签名，相较于 OAuth 2.0 协议更适合用于用户身份鉴别.

一个典型的身份令牌携带的信息如下所示：

{

"iss"："https://idp. example. com"，Token 发布者的身份标识

"sub"："24400320"，Token 对应的用户的身份标识

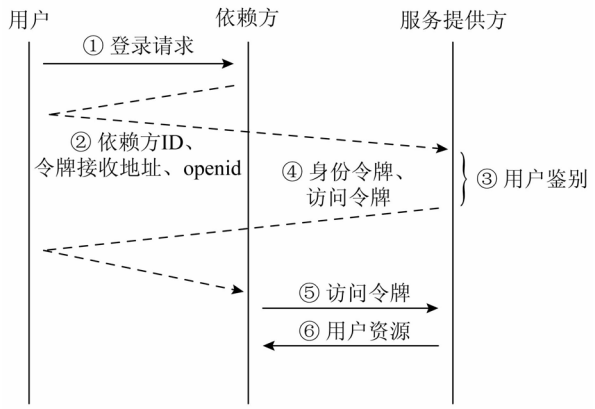
"aud"："s6BhdRkqt3"，Token 对应的依赖方的身份标识

"nonce"："n-0S6_WzA2Mj"，随机数

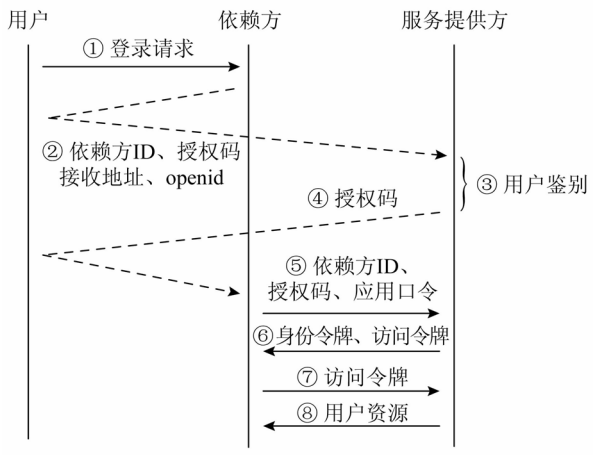
"exp"：1311281970，Token 失效时间

"iat"：1311280970，Token 签发时间

}



(a) 隐式模式



(b) 授权码模式

图 2　OpenID-Connect 协议流程

1.3　SAML 协议

SAML 协议[5]是一个基于 XML 标准，用于在不同的主体之间交换认证和授权信息的数据格式. SAML 的断言是一段描述了用户信息的 XML 语句. 在数据交换的过程中，SAML 通过断言对某

一事实进行声明，分为鉴别声明、属性声明和授权决定声明. 一般鉴别声明用于单点登录的实现.

SAML 用于单点登录场景与 OAuth 2.0 和 OpenID-Connect 相似，只是传输的内容是 XML 文件.

2 单点登录系统的安全假设

在单点登录系统中，身份鉴别凭据在使用以及传输的过程中必须满足以下 2 个特性：完整性和机密性. 依赖方接收的身份鉴别凭据必须是对应用户授权的由服务提供方颁发的凭据，并且用户凭据在任何情况下都不会泄露给用户和依赖方以外的第三者.

IETF 对 OAuth 2.0 协议面临的威胁以及需要满足的安全保证作了详细的描述[6]. 陈君等人[7]也对基于 OAuth 的单点登录系统的安全性进行了分析和评估.

2.1 威胁模型

2.1.1 攻击者目标

在单点登录系统中，我们认为攻击者的目的在于操作受害者在依赖方的账户或者获得受害者的隐私数据，所以攻击者的攻击目标可以归纳为以下 3 条，攻击者实现其中任何 1 条都视为对受害者攻击的成功：

1）以受害者的身份登录依赖方；

2）使受害者以攻击者的身份登录依赖方；

3）在用户不知情的情况下搜集用户数据.

2.1.2 单点登录系统

正常的单点登录场景如图 3 所示：单点登录系统包含 4 种角色：用户代理、服务提供方、依赖方和用户. 用户通过用户代理参与到单点登录过程中，一般在单点登录的协议流程中可以视为一个角色，只在具体的攻击场景中进行详细的区分.

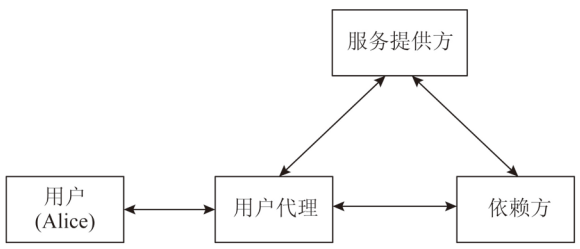


图 3 单点登录系统

单点登录系统的流程如图 4 所示，可以抽象概括如下：

① 一般情况下，单点登录过程从用户开始发起，用户首先访问依赖方请求登录；

② 由依赖方构造向服务提供方申请用户身份凭据的请求，作为响应发送给用户，再由用户转发给服务提供方；

③ 服务提供方验证用户身份；

④ 服务提供方验证用户身份后将身份凭据经过用户代理转发给依赖方；

⑤ 依赖方通过验证身份凭据对应的用户身份（例如，OpenID-Connect 中依赖方通过获得身份令牌中的用户身份并对服务提供方签名进行验证来完成用户身份验证）完成用户登录.

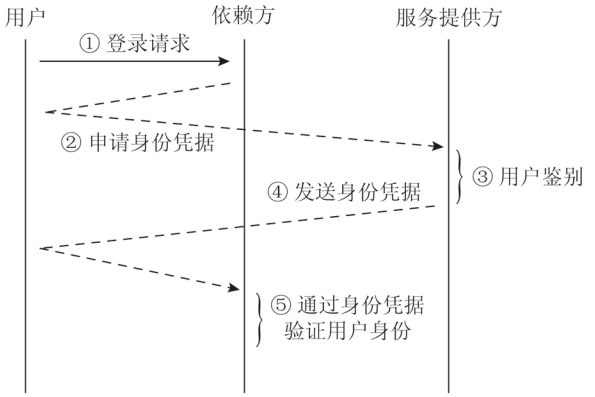


图 4 单点登录协议流程

2.1.3 攻击场景与攻击者能力

根据攻击者在单点登录系统中所扮演角色的不同，攻击者能够拥有不同程度的能力. 由于服务提供方通常是由权威的厂商充当，提供整个单点登录系统的框架，所以一般不考虑服务提供方的身份是攻击者的场景，但是服务提供方在实现单点登录协议时往往会存在漏洞，这些漏洞可能被攻击者在不同的攻击场景下利用. 由于单点登录系统中除了服务提供方之外还有其他 3 种角色可能由攻击者扮演，以及攻击者可以在单点登录系统之外对其进行攻击，所以攻击者的状况大致可以归纳为 4 种不同的场景：

1）场景 1. 攻击者作为用户存在. 攻击者能够获得任意从服务提供方、依赖方发送给用户代理的数据，能够篡改任意由用户代理转发的数据，能够构造任意的对服务提供方、依赖方的请求.

2）场景 2. 攻击者作为依赖方存在. 攻击者能够构造任意的依赖方对用户代理的响应，能够构造任意的依赖方对服务提供方的请求，能够获得用户上传的身份凭据，能够获得用户注册在服务提供方的身份信息.

3）场景 3. 攻击者利用存在安全问题的用户代理. 如果用户代理存在安全问题（针对不同平台可能存在不同的安全问题），攻击者能够窃取和篡改单点登录过程中经过用户代理的数据.

4）场景 4. 攻击者在服务提供方、依赖方、用户代理的网络出口进行监听. 攻击者可以获得单点登录过程中所有的网络数据.

2.2 安全假设

由于以上所描述的攻击场景的存在，单点登录系统必须针对特定的攻击场景提出对应的安全保证来确保单点登录过程的安全性：

攻击场景 1 破坏了单点登录系统中依赖方、服务提供方对用户的信任，所以针对这种场景需要满足：经过用户代理的数据无法被篡改；鉴别过程中的关键步骤（例如 OAuth 2.0 授权码模式中使用授权码换取访问令牌的过程）不能在用户代理实现.

攻击场景 2 破坏了单点登录系统中依赖方的安全性，所以在这种场景下需要满足：某个依赖方获得的所有数据只有在该依赖方是有效的；依赖方不能绕过用户偷偷获取用户的数据.

攻击场景 3 对单点登录系统的网络传输过程提出了安全需求：所有的网络传输过程必须是安全的.

攻击场景 4 对用户代理的安全性以及功能提出了要求：用户代理一定是安全的，不能被攻击者控制；用户代理一定能安全地处理单点登录过程中数据转发的过程.

根据以上针对不同的攻击场景所需要的安全保证，可以将这些内容总结归纳为以下 7 条安全假设：

1）被鉴别用户在依赖方鉴别用户结束之前是不可信的.

根据场景 1 描述的情况，等待验证身份的用户真实身份可能为攻击者，在这种情况下，一旦依赖方将本该在服务器验证的部分下放到用户代理来实现，由于攻击者对于用户代理有着绝对的控制权，那么攻击者就有机会绕过验证的过程，完成攻击目标 1.

2）用户身份凭据一定只能发送给用户和对应的依赖方，并且传输过程一定是安全的.

根据场景 2 的描述，如果服务提供方不能保证只发送给用户和对应的依赖方，那么攻击者有可能作为恶意的依赖方引诱受害者将自己在其他依赖方使用的身份凭据发送给攻击者，这样攻击者便可以实现攻击目标 1. 此外，根据场景 3，单点登录过程中的网络数据一定要经过保护，防止攻击者窃取或者篡改身份凭据等数据，实现攻击目标 1、目标 2 和目标 3.

3）所有的数据一旦在传输或者转发的过程中被篡改，那么鉴别过程一定无法正确完成.

根据场景 1、场景 3 和场景 4，若是用户作为攻击者篡改了服务提供方与依赖方之间转发的数据，如果缺少了对于数据完整性的验证，那么攻击者极有可能实现攻击目标 1.

4）恰当的用户身份凭据一定是与用户、依赖方绑定的私密信息.

如果身份凭据不是私密信息，那么根据场景 1，攻击者便可以用搜集到的受害者的身份凭据信息来伪装成受害者登录依赖方. 另外，根据场景 1 与场景 2 的联合，如果身份凭据没有与用户、依赖方绑定，用户登录依赖方 a 向服务提供方申请获得的身份数据，也可以被攻击者用来在依赖方 b 以用户的身份进行登录，这样攻击者可以通过场景 2 获得受害者的身份凭据，通过场景 1 实现攻击目标 1.

5）用户使用的用户代理一定不会向第三方泄露任何传输的数据.

根据场景 4，如果用户的用户代理会向攻击者泄露传输的数据，那么攻击者很有可能获得用户的身份凭据等数据，实现攻击目标 1.

6）用户使用的用户代理一定要提供安全的消息转发机制，用来完成服务提供方与依赖方之间的消息传递.

在某些平台中，用户代理可能并不只是一个单一的应用，而是由多个应用组合完成. 由于单点登录过程中需要用户代理对服务提供方与依赖方

的数据进行转发，根据场景4，如果消息转发的过程中出现问题，那么攻击者很有可能获得或者篡改用户的身份凭据等数据，实现攻击目标1和目标2.

7）用户一定要有明确的授权行为.

根据场景2，如果单点登录过程缺少明确的用户授权过程，那么攻击者就可以作为恶意依赖方在用户不知情的情况下搜集用户的身份信息.

2.3 密码学上的安全保证

为了保证重要数据的机密性和完整性，在密码学上一般采用加密和签名2种手段完成.

2.3.1 加密

在单点登录系统中，所有关键数据的传输都必须通过HTTPS来实现，通过加密的方式保证任何数据在传输数据的两端之间的通信过程中不会被第三方获取.

2.3.2 签名

在单点登录系统中，由于用户是不可信的，同时浏览器的重定向机制使得许多关键数据在用户端是以明文出现的，所以必须要保证这些信息在经过用户端的过程中不会被篡改，所以此处关键数据的保护要通过服务器对其签名来保证. 缺少签名或者签名被绕过导致的安全问题我们将在后文进行较为详细的描述.

此外，许多单点登录协议中的身份鉴别凭据要通过签名保证其正确性. 例如OpenID-Connect中的身份令牌就是先将身份鉴别的关键信息，包括服务提供方的身份、依赖方的身份、用户的身份等信息通过Base64编码，然后经过指定的签名算法进行签名，发送给依赖方，依赖方便可以通过对签名的验证来确定身份令牌中数据的准确性.

3 单点登录协议的安全分析

现实场景中单点登录系统的设计与实现往往存在着许多的问题. 有大量的研究成果披露了实际的单点登录系统中存在的漏洞，这些漏洞往往是由于系统的设计与实现中违背了上文描述的单点登录协议的安全假设中的1条或者多条造成的. 本节将对实际实现中存在的漏洞与其违背的安全假设进行详细的描述，内容如表1所示：

表1 现实单点登录系统存在的问题

单点登录系统安全问题	违背的安全假设
关键信息签名被绕过	假设1,3
对协议中凭据的误用	假设3,4
缺少机密性保护	假设2,7
移动端单点登录系统面临的问题	假设5,6
依赖方开发者对于协议的错误应用	假设1

3.1 关键信息签名被绕过

无论在何种协议中，对关键数据的签名是防止数据被篡改最简单有效的手段之一，但是在实际情况下，往往存在着关键数据的签名过程或者验签过程被绕过的情况[8-9].

在Google ID中，依赖方可以通过用户重定向至服务提供方一个请求，记为$Request_{RP}$，其中包含1个参数列表，其中记录了向服务提供方请求的数据，例如UserName, Email. 之后由服务提供方重定向至依赖方的响应，记为$Response_{IdP}$，其中包含与请求相同的参数列表，以及列表中请求数据的值与签名，这些信息，例如Email常被依赖方用作身份标识.

由于请求与响应都要经过用户浏览器，那么对于其中未经签名的数据用户都可以进行篡改. 在这个例子中，攻击者可以以自己的身份在服务提供方登录（设为Bob），在依赖方请求单点登录，并将$Request_{RP}$的参数列表进行篡改，例如删去Email，这样$Response_{IdP}$中的参数列表中也不会包含Email，以及签名中也不会包含Email，此时只要在$Response_{IdP}$通过浏览器时在最后拼接受害者的Email（例如alice@a.com），因为该值不存在于$Response_{IdP}$参数列表中，依赖方就不会对其进行验签，也就是会接受任意的Email值，那么此时如果Email被用于依赖方的身份鉴别，那么攻击者便能够以受害者的身份登录依赖方.

在上述的场景中，关键的数据在经过被鉴别身份的用户时未经保护，导致其可以被篡改，从而产生漏洞，违背了单点协议安全假设的第1条与第3条.

3.2 对协议中凭据的误用

单点登录协议的使用过程中需要注意的一个问题就是协议本身的安全假设与使用场景安全假设的背离. 例如OAuth 2.0协议本身作为授权协

议而设计，由于其目的并不是用于身份鉴别，所以并不满足单点登录协议的安全假设4，访问令牌并没有与任何依赖方进行绑定. 这样一旦任何攻击者注册为合法的依赖方，那么任何在该依赖方登录的用户就会将与自己身份绑定的访问令牌发送给攻击者，攻击者便可以使用该令牌以该用户的身份登录其他依赖方[9-11]. 同时由于在传输过程中访问令牌被篡改后无法被识别而终止身份鉴别过程，也不能满足第3条安全假设.

在实际环境中，OAuth 2.0在很多时候被当作身份鉴别协议使用，那么必须额外设计添加访问令牌与相对应的依赖方之间的关联，这种设计常通过以下2种方式实现：

1）由依赖方直接向服务提供方获取Token；

2）额外对依赖方和Token进行绑定.

OAuth 2.0的授权码模式就是采用了第1种方式，实际上使用了与依赖方绑定的授权码作为用户凭据，访问令牌的获取通过依赖方服务器与服务提供方服务器之间的直接通信来获得，并通过双方共享的应用口令认证的方式来保证依赖方与访问令牌的绑定，同时这一过程还通过HTTPS来保证传递的访问令牌不被篡改. 还有许多服务提供方使用第2种方式保证访问令牌的正确性，如Facebook等[12]，例如将令牌与用户ID进行绑定，同时使同一个用户在不同的依赖方拥有不同的用户ID，实现了依赖方与令牌的绑定. 但是在协议的具体实现中这2种方式往往存在着问题：部分依赖方开发商对于OAuth 2.0的授权码模式的理解不够，将本应该在依赖方服务器实现的获取Token的过程放在客户端实现；一些服务提供方开发商在使用访问令牌与用户ID的过程中，对每个用户使用了唯一的用户ID，没有实现令牌与依赖方的绑定. 这2种情况都导致了令牌有可能被攻击者篡改，依赖方无法保证请求登录的用户身份的正确性.

3.3 缺少机密性保护

在讨论这个问题时我们首先作出以下的假设：

1）整个单点登录系统中所有与凭据传输的过程都通过HTTPS实现，也就是凭据不会在传输过程中被泄露；

2）用户只会使用正规浏览器厂商提供的浏览器，浏览器是可信的，浏览器不会泄露用户凭据.

在不考虑单点登录系统中采用的技术出现漏洞的情况下，从协议设计角度进行分析，服务提供方需要避免以任何形式将凭据发送给攻击者. 在实际的单点登录系统中，服务提供方会通过浏览器重定向的方式将关键的用户凭据发送给依赖方，如OAuth 2.0授权码模式中的授权码或者身份令牌，此时如果服务提供方错误地将重定向地址设置为攻击者的地址（攻击者诱导用户访问恶意的地址），那么用户凭据就会泄露给攻击者[8]，在这种场景下，单点登录协议安全性假设的第2条没有得到满足. 所以在单点登录系统的设计中，必须将依赖方身份与用户凭据的接收地址进行绑定，避免将用户凭据发送给攻击者.

此外，单点登录系统中必须要满足第7条安全假设：在用户授权给某个依赖方时必须明确且醒目地要求用户进行确认[10,13]. 在上一部分的描述中介绍了如果攻击者诱导用户访问恶意地址来窃取用户的鉴别凭据，如果在这个过程中明确地要求用户对该项操作进行授权，那么用户就有机会发现请求授权的主体与现在自己访问的地址的所有者不同，便可以拒绝此次授权以防止用户凭据泄露. 另外，在单点登录系统中，许多用户凭据并不仅仅用于依赖方鉴别用户身份，还可以向服务提供方请求用户信息，如用户名、Email等. 所以在缺少用户明确授权的情况下，攻击者如果在服务提供方注册为合法依赖方，就有机会在用户没有察觉的情况下搜集用户信息.

3.4 移动端单点登录系统面临的问题

随着移动智能设备的普及，现阶段单点登录系统已经实现了由传统的B\S模式向移动端的转变. 移动端的单点登录最主要的问题是缺少在传统方式中作为可信基的浏览器，那么就无法用使用浏览器的重定向模式来保证单点登录的安全性[10].

移动端单点登录为了配合已有的协议，一般采用2种方式在移动端模拟浏览器的重定向：

1）在依赖方应用中使用WebView来完成重定向；

2）使用APP间的消息传递机制来代替重定向.

对于使用WebView的情况，由于WebView都是内嵌于依赖方的APP中，这种情况下攻击者简单地在WebView中注入JavaScript代码实现对用户的攻击[14-15]，不能满足单点登录协议安全

假设的第5条，所以使用WebView实现单点登录是不安全的.

使用应用间消息传递机制实现重定向最主要的问题是在现有的移动端操作系统中，应用间消息传递时无法对接收方以及发送方进行身份鉴别，所以无法确定消息是正确地发送以及正确地接收[16-17]，无法满足单点登录协议安全假设中的第6条. 虽然在现在的Android系统中，单点登录系统通过Android系统提供的startActivityForResult机制实现了对依赖方应用的身份鉴别，但是却没有对服务提供方应用的身份鉴别[10]. 在这种情况下，如果存在恶意的应用冒充服务提供方响应依赖方应用的请求，并返回了攻击者的用户凭据，那么就有可能使受害者在未察觉的情况下在依赖方应用登录攻击者的账号. 如果该应用是网盘类应用受害者极有可能将隐私信息泄露给攻击者.

3.5 依赖方开发者对于协议的错误应用

在单点登录系统中依赖方的开发者可能对协议理解不足、违背了身份服务提供方默认的安全假设，其中最为常见的问题就是依赖方在客户端实现本应该在服务端实现的流程.

例如在OAuth 2.0协议的授权码模式中，步骤⑤，⑥的授权码换取访问令牌的过程必须要在依赖方的服务端实现，否则不仅访问令牌面临被篡改的危险，而且存储在客户端的应用口令很容易被攻击者窃取. 但是由于许多开发者对于OAuth 2.0协议的理解不足，把换取令牌的过程放在客户端实现，导致面临着被攻击的风险[9-10,17].

此外，许多开发者对于身份凭据的认识也不足. 单点登录过程中的身份凭据一定是私密信息，例如在OAuth 2.0隐式模式中，由用户上传至服务器作为用户凭据的必须是访问令牌. 然而许多开发者却错误地将可以被攻击者获取的用户公共信息当作身份凭据，例如先使用访问令牌换取用户信息，之后再使用用户信息中的数据，如Email作为用户凭据. 因为用户信息Email等很多情况下是公开信息，所以攻击者可以伪造任意用户身份登录依赖方[17-18].

在以上2种场景下，开发者错误地将本应该在服务端执行的逻辑放在了客户端，违背了单点登录协议安全假设的第1条.

4 结　　语

单点登录系统的安全性保证是目前单点登录研究的一个重要方向，本文通过对于各种单点登录协议的分析，提炼出了统一的单点登录协议流程，并根据实际单点登录系统中面临的安全威胁，总结出了相应的单点登录系统需要满足的安全假设. 最后，对现在已经发现的单点登录系统漏洞进行分析，发现这些漏洞产生的根本原因是违背了1条或者多条安全假设，从而为后续单点登录系统的设计和安全性分析奠定了基础.

参 考 文 献

[1] Hardt D. The OAuth 2.0 Authorization Framework [S]. Fremont: Internet Engineering Task Force (IETF), 2012

[2] Jones M. The OAuth 2.0 Authorization Framework: Bearer Token Usage [S]. Fremont: Internet Engineering Task Force (IETF), 2012

[3] Sakimura N, Bradley J, Jones M, et al. OpenID Connect Core 1.0 Incorporating Errata Set 1 [S]. San Ramon: OpenID Foundation (OIDF), 2014

[4] Jones M. JSON Web Token (JWT) [S]. Fremont: Internet Engineering Task Force (IETF), 2015

[5] Cahill Conor P, Hughes J, Lockhart H, et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 [S]. Burlington: Organization for the Advancement of Structured Information Standards (OASIS), 2005

[6] Lodderstedt T. OAuth 2.0 Threat Model and Security Considerations [S]. Fremont: Internet Engineering Task Force (IETF), 2013

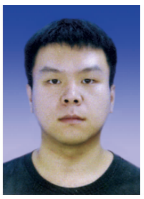
[7] 陈君，张生. 基于OAuth单点登录系统的安全性分析和评估[J]. 电子科技，2017，30(9):165-168

[8] Wang R, Chen S, Wang X, et al. Signing me onto your accounts through Facebook and Google: A traffic-guided security study of commercially deployed single-sign-on Web services [C] //Proc of Symp on Security and Privacy. Piscataway, NJ: IEEE, 2012: 365-379

[9] Zhou Y, Evans D. SSOScan: Automated testing of Web applications for single sign-on vulnerabilities [C] //Proc of USENIX Security Symp. Berkeley, CA: USENIX Association, 2014: 495-510

[10] Chen E Y, Pei Y, Chen S, et al. OAuth demystified for mobile application developers [C] //Proc of Conf on Computer and Communications Security. New York: ACM, 2014: 892-903

[11] Wang H, Zhang Y, Li J, et al. The achilles heel of OAuth: A multi-platform study of OAuth-based authentication [C] //Proc of Annual Computer Security Applications Conf. Piscataway, NJ: IEEE, 2016: 167-176
[12] Facebook. Manually build a login flow [EB/OL]. [2018-12-11]. https://developers. facebook. com/docs/facebook-login/manually-build-a-login-flow
[13] Yang R, Li G, Lau W C, et al. Model-based security testing: An empirical study on OAuth 2.0 implementations [C] //Proc of Computer and Communications Security. Piscataway, NJ: IEEE, 2016: 651-662
[14] Luo T, Hao H, Du W, et al. Attacks on WebView in the Android system [C] //Proc of Computer Security Applications Conf. New York: ACM, 2011: 343-352
[15] Mohsen F, Shehab M. Hardening the OAuth-WebView implementations in Android applications by re-factoring the chromium library [C] //Proc of Int Conf on Collaboration and Internet Computing. Piscataway, NJ: IEEE, 2017
[16] Wang R, Xing L, Wang X F, et al. Unauthorized origin crossing on mobile platforms: threats and mitigation [C] //Proc of ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 635-646
[17] Yang R, Lau W C, Shi S. Breaking and fixing mobile APP authentication with OAuth 2.0-based protocols [C] //Proc of Applied Cryptography and Network Security. New York: Springer, 2017: 313-335
[18] Wang H, Zhang Y, Li J, et al. Vulnerability assessment of OAuth implementations in Android applications [C] //Proc of Computer Security Applications Conf. New York: ACM, 2015: 61-70



郭丞乾

博士研究生，主要研究方向为密码工程与应用.

guocqian@gmail. com

蔡权伟

博士，助理研究员，主要研究方向为网络服务安全.

caiquanwei@iie. ac. cn

林璟锵

博士，研究员，主要研究方向为应用密码学、数据安全与隐私、网络与系统安全.

linjq@is. ac. cn

刘丽敏

博士，副研究员，主要研究方向为应用密码学、网络与系统安全.

liulimin@iee. ac. cn