文章编号:0253-2778(2019)07-0564-08

基于OAuth2.0协议的智慧校园认证系统研究

高保忠[1], 杜首燕[2], 李信治[3], 王新华[1]

(1. 山东师范大学信息科学与工程学院, 山东济南 250358; 2. 中国科学院计算机网络信息中心, 北京 100190;
3. 云南大学信息学院, 云南昆明 650500)

摘要: 以山东师范大学智慧校园建设为切入点,基于OAuth2.0协议设计并实现智慧校园认证系统. 首先介绍了OAuth2.0认证授权技术的系统角色和授权流程;然后分析了智慧校园认证系统的设计和具体实现功能;最后通过理论分析和流程实验对系统进行测试,改善了校园数据获取的安全性和可靠性.

关键词: OAuth2.0协议; 智慧校园;用户认证和授权;开放平台;开放API

中图分类号: TP391 **文献标识码**: A doi: 10.3969/j.issn.0253-2778.2019.07.006

引用格式: 高保忠, 杜首燕,李信治,等. 基于OAuth2.0协议的智慧校园认证系统研究[J]. 中国科学技术大学学报,2019,49(7):564-571.

GAO Baozhong, DU Shouyan, LI Xinzhi, et al. Research on an OAuth2.0-based unified authentication system in the smart campus environment[J]. Journal of University of Science and Technology of China, 2019,49(7):564-571.

Research on an OAuth2.0-based unified authentication system in the smart campus environment

GAO Baozhong[1], DU Shouyan[2], LI Xinzhi[3], WANG Xinhua[1,*]

(1. *Institute of Information Science and Technology, Shandong Normal University, Jinan*, 250358
2. *Computer Network Information Center, Chinese Academy of Science, Beijing*, 100190
3. *School of Information Science and Engineering, Yunnan University, Kunming*, 650500)

Abstract: Based on the construction of Shandong Normal University's smart campus system, this paper summarizes the research methods of building the smart campus authentication system, which introduces the system role and authorization procedure of OAuth2.0-Based authentication and authorization technology, and analyzes the concrete implementation of the smart campus authentication platform. By conducting security experiments and theoretical analysis, the security and reliability of campus data acquisition has been improved.

Key words: OAuth2.0; smart campus; authentication and authorization; open platform; open API

0 引言

近年来,随着"互联网+"的概念的迅速普及以及Web技术的广泛应用,大多数校园的不同部门都各自拥有独立的信息系统. 各信息系统内,存放着大量格式不统一的师生信息. 在现有环境下,对师生情

收稿日期: 2018-06-20; **修回日期**: 2018-09-18

基金项目: 国家自然科学基金(90612003, 61602282),中国博士后科学基金(2016M602181),山东省自然科学基金(ZR2013FM008),山东省科技发展计划项目(2011GGH20123)资助.

作者简介: 高宝忠, 男, 1981年生, 硕士/讲师. 研究方向: 智慧为城市. E-mail: gaobaozhong@sdnu.edu.cn

通讯作者: 王新华, 博士/教授. E-mail: wangxinhua@sdnu.edu.cn

况的总体分析、对师生的完整信息描述逐渐成为校园信息化的瓶颈，智慧校园的实现迫切需要一个统一的、安全可靠的信息发放平台.

由于传统的 HTTP 认证方式必须提供用户凭证，而且在认证完成后，拥有用户许可令牌就意味着可以获取用户所有的受保护的数据，因此传统的 HTTP 认证方式已不满足当下的单一用户凭证许可，向第三方平台共享有限的受保护数据的需求. 本文从山东师范大学的实际情况出发，基于 OAuth2.0 协议设计并实现智慧校园认证系统，使得用户不必提供第三方认证凭证，在使用第三方代理登录目标服务器的情况下，通过授权允许第三方获取指定的信息，具有简单方便、认证流程安全可靠、对数据的发放可控等诸多优点. 在统一认证的支撑下，校园智慧应用所接入的所有第三方平台都是一个互联的共享整体，进一步改善了校园数据获取的安全性和可靠性，推动了校园信息化建设，使校园网络更好地服务于学校事务以及师生生活.

1　OAuth2.0 认证授权技术

1.1　协议介绍

OAuth(open authorization)即开放授权协议，提供了一个安全、可靠的框架供第三方应用在一定授权和限制下访问 http 服务[1]. 智慧山师使用 OAuth 版本为 OAuth2.0 及 OAuth 1.0a，目前向全校师生开放 OAuth2.0 接口. 获取用户授权分为 4 种方式：

(Ⅰ)授权码授权(authentication code grant)：客户端是 Web 服务器的一部分，通过 http 请求实现，是 OAuth1.0 流程的简化版本.

(Ⅱ)隐式授权(implicit grant)：客户端运行于用户代理内(通常用 JavaScript 等脚本语言在浏览器中实现).

(Ⅲ)资源所有者密码凭证授权(resource owner password credentials grant)：这种授权许可的类型需要最终用户和客户端有很强的信任关系，客户端可以直接使用资源拥有者的私有证书(用户名和密码)用作访问许可获取 access token.

(Ⅳ)客户端凭证授权(client credentials grant)：客户端使用它的私有证书(client_id 和 client_secret)去获取 access token[2].

OAuth2.0 认证协议的参与实体通常包括 7 种：① RO(resource owner)：能够对受保护资源进行访问许可控制的实体，相当于 OAuth1.0 中定义的 User. ② RS(resource server)：存储用户的数据资源，能够接受和响应受保护资源访问请求的服务器. ③ Client：获取授权和发送受保护资源请求的第三方应用，相当于 OAuth1.0 中定义的 Consumer[3]. ④ AS(authorization server)：能够成功认证资源拥有者身份和获取授权，并发放 access token 的服务器. ⑤ user-agent：适合所有无 server 端配合的客户端，一般为用户浏览器. ⑥ access token：授权服务器创建的数据，使得资源服务器通过客户端请求[4]. ⑦ authentication code：授权服务器可以检查的一段数据，在事务请求阶段使用[4].

1.2　协议流程

在 OAuth2.0 协议流的定义下，第三方获取资源必须按照获取认证、获取访问资源令牌、通过令牌获取指定资源的顺序[5]，该授权流程主要分为 6 个步骤，如图 1 所示.

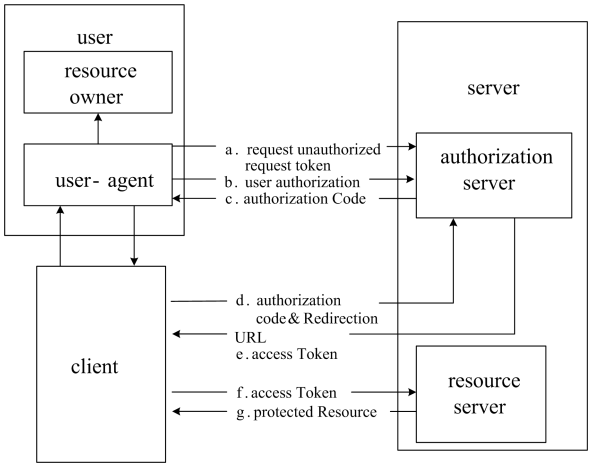

图 1　OAuth2.0 协议流程

Fig. 1　OAuth2.0 protocol process

(Ⅰ)client 向 RO 发送授权请求，其中包含类型，client Id 和重定向 URL，开始用户授权流程.

(Ⅱ)RO 同意授权，并返回 AC 给 UA.

(Ⅲ)用户完成授权后，client 使用 AC 和重定向 URL 向 AS 请求 AT.

(Ⅳ)AS 认证 client 和 AC 的有效性. 如果 client 用户的个人认证和接入权限通过且有效，client 可以得到 AS 分发的 AT.

(Ⅴ)client 通过 AT 向 RS 请求受保护资源.

(Ⅵ)RS 验证 AT 的有效性，验证成功后响应资源请求，client 获得权限通过指定信息端口得到受保护资源.

协议流程为：首先第三方用户申请访问资源服

务器,资源服务器生成 pre-token 后将其发送给客户端,同时同步给授权服务器;然后第三方应用访问授权服务器的安全节点,授权服务器验证 pre-token,如果验证成功则继续,否则授权失败;最后第三方应用访问授权节点申请 authorization code,使用申请到的 authorization code 换取 access token. 当第三方应用获取到 access token 后,向资源服务器申请资源时出示 access token 和资源服务器生成的 pre-token;如果验证失败则授权失败,否则授权成功[6].

1.3 Open API

API 是软件模块的设计者指定它在已发布的 API 文档中的 API,无需对其提供的功能做任何判断[7]. 在线服务拥有的信息以公共应用编程接口(API)的形式提供给第三方应用,通常使用 http[8] 作为通信协议并且依赖于 RE 呈现状态传输(REST)体系结构风格[9]. 智慧校园认证系统的接口 API 工作流程如下:

(Ⅰ)前提说明

①智慧校园认证系统已经开通了该 Open API 的使用权限. 从 API 列表的接口列表中可以看到,有的接口是完全开放的,有的接口则需要提前提交申请,以获取访问权限.

②准备访问的资源是用户授权可访问的. 用户调用该 Open API 读写某个 Open ID(用户)的信息时,必须是该用户已经对智慧校园认证系统进行了该 Open API 的授权.

③已经成功获取到 AT,并且 AT 在有效期内.

(Ⅱ)调用 Open API 接口

网站需要将请求发送到某个具体的 Open API 接口,以访问或修改用户数据. 调用所有 Open API 时,除了各接口私有的参数外,所有 Open API 都需要传入如表 1 所示的基于 OAuth2.0 协议的通用参数[10].

表 1 通用参数

Tab. 1 General parameters

access_token	可通过使用 Authorization_Code 获取 AT 来获取. AT 有 3 个月有效期.
oauth_consumer_key	第三方应用登录成功后,分配给应用的 appid.
openid	用户的 ID,与智慧山师账号对应.

以图书馆借阅信息获取接口为例,接口权限是 read_personal_library,功能是返回当前登录用户图书馆指定时间段内的借阅数量及借阅信息集合.

请求方式:

GET

http://i.sdnu.edu.cn/oauth/rest/library/getborrowlist 所需参数如表 2 所示.

表 2 图书馆借阅信息获取接口参数

Tab. 2 Library obtains borrowing information interface parameters

请求字段名称	必选	字段类型	字段说明
start	false	string	开始日期 (formal as yyyy-MM-dd HH:mm:ss)
end	false	string	结束日期(格式同上)
count	false	int32	返回数量 (1—50, 默认为 10)
index	false	int32	返回页码 (默认为 1)

成功返回后,可获取用户数据.

```
[
    {
        "identityNumber":"2013001001",
        "bookName":"书名",
         "borrowDate":"2014-01-01T08:00:
00",
        "mustReturnDate":"2014-01-31T08:
00:00",
        "isRenew":false
    },
    {
其他同上
    }
]
```

在基于 OAuth2.0 的智慧校园认证系统中,我们可以通过个人基本信息接口,个人一卡通信息接口,个人图书馆信息接口,失物招领信息接口,学校公共信息接口,学校新闻信息接口以及天气信息接口来获取所需要的信息. 通过分析 3 年的接口访问次数,我们得出访问次数最多的三个接口为图书馆、一卡通和用户个人信息 API,从图 2 2014—2016 年变化趋势图可知,一卡通和用户个人信息 API 近年较稳定,图书馆 API 访问次数 2015 年较 2014 年明显增加,后趋于稳定,说明我校新图书馆的建设有效提高了在校师生的学习积极性,对我校学风建设具有显著的促进意义.

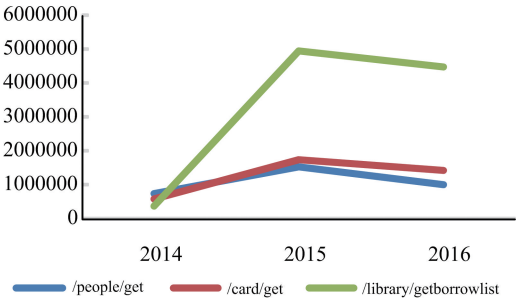

图 2　2014—2016 年用户、一卡通、图书馆 API 访问次数变化趋势图

Fig. 2　Trends in the number of API visits by users, cards and libraries in 2014—2016

2　基于 OAuth2.0 的智慧校园认证系统设计

2.1　认证系统整体架构

本文以山东师范大学智慧校园建设为切入点，基于 OAuth2.0 协议，设计开发智慧校园开放认证系统，架构采用的是分层架构的设计理念，将系统的总体架构分为基础设施层、应用服务层以及信息提供层，其各层的主要功能如下：

（Ⅰ）基础设施层：主要工作是及时准确地收集处理各种信息，通过 RFID 识别、红外感应器、视频采集、GPS 等技术和设备对校园信息进行采集和动态监控，安全无误地将从硬件设备采集到的信息传送到数据层.

（Ⅱ）应用服务层：应用层数据的主要工作是有效地整合和管理各种信息，实现信息的统一管理. 基于现有的各项管理系统，例如财务管理系统、学生管理系统、教工管理系统、科研管理系统、设备管理系统、后勤管理系统等，利用云计算、云存储的技术，提供统一的管理平台，便于第三方应用调用开发，应用服务架构如图 3 所示.

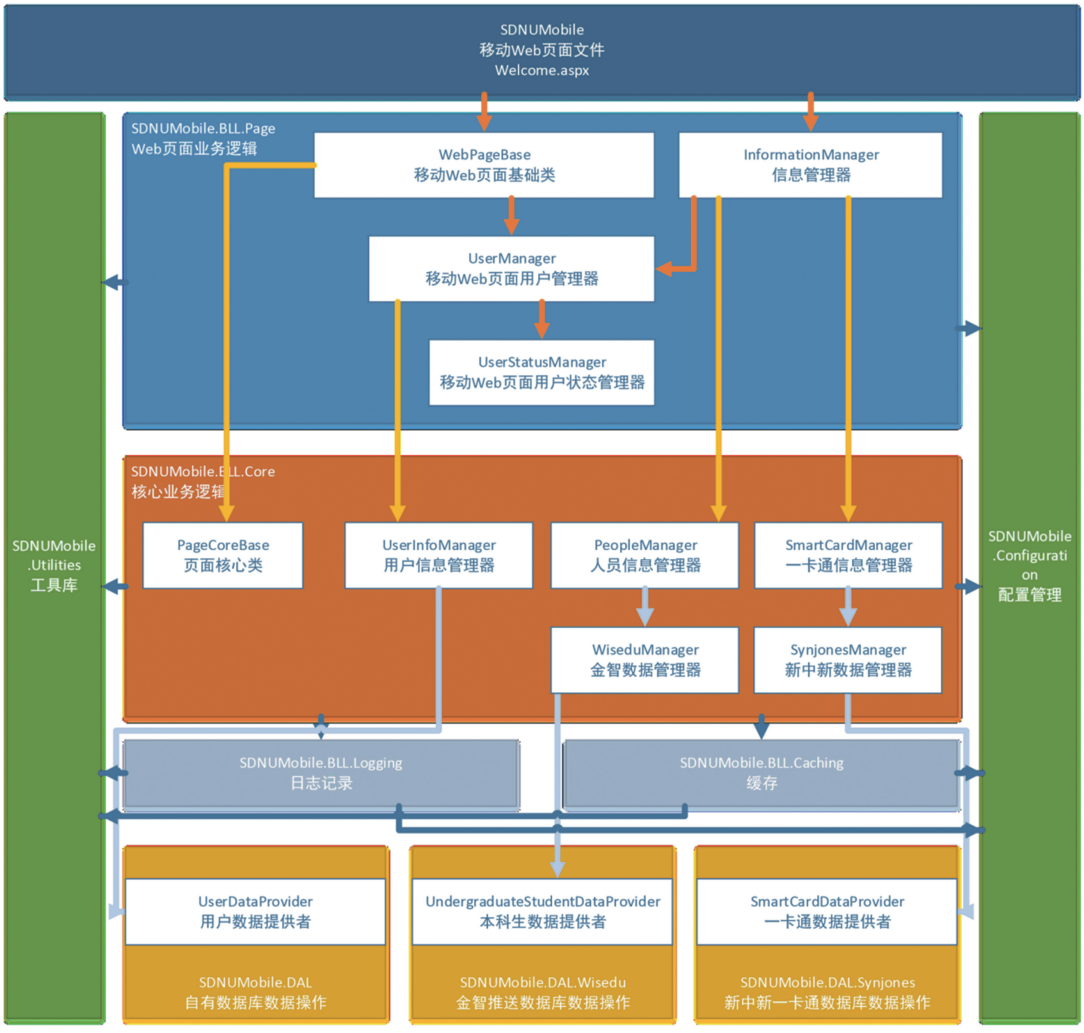

图 3　智慧校园应用服务架构图

Fig. 3　Smart campus application service architecture

(Ⅲ)信息提供层：信息提供层的主要工作是为师生提供具体、有效地服务的平台. 在这个平台上，按照统一的数据规范标准，通过身份认证开放平台的学校信息和教师、学生信息，师生可以使用共享平台上的教学资源和科研资源，形成一个互联的共享整体.

2.2 认证系统工作流程

智慧校园开放认证系统，工作流程具体分为第三方应用获取临时凭证、令牌凭证，用户同意或拒绝授权第三方应用，用户信息获取包含3个步骤：

(Ⅰ)第三方应用获取临时凭证流程

(a)申请信息附加到请求包：将客户端凭证、后续需要的回调地址以及请求的控制信息(包括防止重播的单次值、请求发出的时间戳等内容)按照认证系统的要求附加到请求包中.

(b)请求参数签名：将http请求方式、http协议请求包的URI、指定的请求键值对排列编码成为基础字符串(base string)，再由HMAC-SHA1算法或RSA-SHA1算法产生消息文摘(即签名)，参与签名算法的密钥是与客户端凭证相对应的密钥.

(c)封装：将签名添加到http请求头中，最终封装成为获取临时凭证的合法请求包.

(d)验证请求合法性：第三方应用通过客户端凭证，向OAuth2.0认证系统发送请求临时凭证的请求包，认证系统在接收到获取临时凭证的请求时首先对请求包进行合法性检查；然后与请求体中的签名相比较，若完全一致，则说明请求合法可信；最后生成临时凭证以及相应的密钥，并返回给第三方应用.

(Ⅱ)用户授权第三方应用流程

(a)临时凭证认证页面：第三方应用通过临时凭证对请求包进行封装，使其包含回调地址，并引导用户进行授权认证，此时网页将会跳转到携带含有临时凭证信息的认证页面. 在该作用域下，可以避免向第三方应用公开认证系统的用户凭证.

(b)用户同意或拒绝授权：若用户许可授权，认证系统将临时凭证激活成为令牌凭证，并跳转到之前传递到认证系统之中的回调地址，第三方应用通过触发回调地址中的事件，使用临时凭证来换取令牌凭证；若用户拒绝授权，则清理相应的临时令牌，并告知第三方应用用户拒绝授权，认证系统将不予发放令牌凭证.

(Ⅲ)用户信息获取流程

(a)第三方应用请求：当第三方应用获得令牌凭证以及相应的密钥后，发起请求时需要有请求签名，以确保请求可信，此时要获取用户的受保护信息，参与签名算法的密钥不仅要包含与客户端凭证匹配的密钥，还要包含与令牌凭证匹配的密钥.

(b)服务器验证：服务器在接受用户信息请求时，需要检验请求包是否合法、请求的Web接口是否存在、发起请求的第三方应用是否具备调用接口的权限、调用Web接口的参数是否合法一系列的验证过程，当全部通过后，方能执行服务器内的业务过程，从而获取相应的受保护的资源. 其中，受保护的资源是在用户许可授权中罗列的项目.

3 系统安全性测试

为检测系统的有效性和安全性，基于OAuth2.0的智慧校园认证系统测试分为设计安全性测试和注入安全性测试两个方面：

3.1 设计安全性测试

(Ⅰ)CSRF攻击测试

以一卡通数据获取接口为例，通过网站正常获取数据，从js中加载JSON数据，访问http//i.sdnu.edu.cn/card/get/info接口，检验cookie登陆状态，并返回JSON数据.

开始尝试CSRF攻击，在同样的cookie条件下尝试访问http//i.sdnu.edu.cn/card/get/info，后台检验请求头中不包含X-Requested-With选项，返回错误：{"status":"error","result":"referer_invalid"}，故应防止攻击导致数据泄露.

若从第三方网站进行攻击，由于网站已经防止跨域攻击，同样无法获取数据.

返回错误：No 'Access-Control-Allow-Origin' header is present on the requested resource.

(Ⅱ)XSS攻击安全性测试

从第三方网站尝试获取数据：

>$.getJSON("http://i.sdnu.edu.cn/card/get/info? 0.3132845529366785");

返回错误：

XMLHttpRequest cannot load http://i.sdnu.edu.cn/card/get/info? Link? url = f6LtFstZ1zpQAAbp9HC4eZtRcoFq-07mpvRYEnDQ0VcSWIm5I9qnd6HM1hhiv7Xmh2008Nd6vExMW9krveVWL: 10.3132845529366785. No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://baike.

baidu.com' is therefore not allowed access.

综上可看出,网站防止跨域请求,通过 XSS 攻击无法获取受保护数据,因此智慧校园认证系统降低了数据泄露的风险.

3.2　注入安全性测试

(Ⅰ)Scope 参数注入测试

请求用户授权时尝试注入 Scope 参数:

https://i.sdnu.edu.cn/oauth/authorize_code?response_type=code&scope='or 1=1'&client_id=00000000000000000000000000000000&oauth_version=2.0.

返回错误:client no permission.

由此说明 Scope 在后台检查调用时,中间加了一层缓存,并不是直接连接数据库,注入字符串被单纯的当作字符串而不是 SQL 语句处理,注入失败.

(Ⅱ)伪装客户端测试

(a)准备申请认证的参数

攻击者知道客户端 ID 并尝试伪装客户端来骗取用户授权和获取数据,所需参数如表 3 所示.

表 3　参数列表

Tab.3　Parameter list

response_type	第一步中固定为 code
client_id	服务端颁发给客户端的凭证 ID
redirect_uri	需要返回 AuthorizeCode 的重定向 URI
scope	本次认证申请的权限范围
oauth_version	请求的认证版本(2.0)

样例:

https://i.sdnu.edu.cn/oauth/authorize_code?response_type=code&scope=BasicAuth,None&client_id=00000000000000000000000000000000&oauth_version=2.0.

(b)用户登录选择授权

引导用户登录:用户点击在第三方应用登录界面上的"使用智慧山师账号登录",第三方应用跳转到智慧校园身份认证开放平台的登录授权页面,输入账号密码后,用户登录成功,同时登陆智慧山师,如图 4 所示.

引导用户授权:用户登录成功后,提示用户是否同意将个人基本信息、一卡通、图书馆、失物招领等个人情况信息授权给第三方应用访问.如图 5 所示,授权后跳转到第一步中的重定向 URI 并附上 authorize code.

图 4　用户登录界面

Fig.4　User login interface

图 5　用户授权界面

Fig.5　User authorization interface

(c)返回授权码

默认回调地址:https://i.sdnu.edu.cn/oauth/?authorize_code=YoCRIJWg932Y1NioNeO2uleyHtVN4Ps5.

(d)用授权码换取 access token

在使用授权码换取 token 时,攻击者并不知道 ClientSecret.

样例:

http://i.sdnu.edu.cn/oauth2/access_token?grant_type=authorization_code&authorize_code=YoCRIJWg932Y1NioNeO2uleyHtVN4Ps5&client_id=00000000000000000000000000000000,所需参数如表 4 所示.

表 4　参数列表

Tab.4　Parameter list

grant_type	此步固定为 authorization_code
authorize_code	上步中获得的授权码
client_id	客户端 ID

返回错误:client no permission.

通过测试结果得知,通过第三方应用应该保证换取 access token 时应该是在服务器进行的,不会被抓取到 secret.

经过一段时间的上线测试,发现 AT 缓存数据库内存放了有效期内的 token,从用户授权认证到获取 token,全部过程遵循 OAuth2.0 协议,通过 https 协议通信,确保了通信过程中的安全性,系统

运行较稳定.

4 系统性能分析

OAuth2.0是一种具有授权功能的协议,用于控制和管理对Web服务的访问[11].采用OAuth2.0实现的智慧校园认证系统,校园智慧应用所接入的所有第三方平台都是一个互联的共享整体,本文从山东师范大学智慧校园建设的实际情况出发,充分考虑目前信息系统和资源整合过程中必须面对的实际问题,设计并实现智慧校园认证系统,并分析该认证系统的功能和实现意义.

(Ⅰ)基础用户数据管理:通过统一的数据规范标准,收集用户验证的主要特征,将分散在教工管理系统、学生管理系统、财务管理系统、科研管理系统、档案管理系统及教务管理系统中的用户进行统一管理,形成包含教师信息、学生信息、学校信息和班级信息的基础数据库,实现不同应用系统之间独立管理和验证用户[12],第三方应用接入商可以通过具有权限的接口批量获取相关信息,开放接口必须具有跨平台性和兼容性[13],使得用户和第三方应用以安全可信的方式进行访问,解决了应用系统之间用户数据同步和更新问题.

(Ⅱ)统一身份认证:采用OAuth2.0协议,在用户授权的整个过程中,用户不必向第三方平台透露用户名、密码等信息,授权后的http通信中以数字签名和访问令牌access token取代用户信息,避免了用户敏感信息的泄露.用户中心提供统一的账号管理和操作,用户信息通过消息总线传递给应用系统,各应用无需开发用户管理,可以将接收的账号信息存入本地,保留应用系统原有权限配置功能[14],同时OAuth2.0可以与PKI(公钥基础设施)相结合,以丰富认证功能,解决了身份认证、授权问题以及云环境开放架构下的用户资源安全问题.

(Ⅲ)跨平台资源共享:资源共享平台存储的只是资源的地址信息以及资源令牌,并没有存储资源本身,用户可以通过共享平台获取资源令牌,从而跨平台的进行安全可靠的资源共享,在一体化资源架构OAuth2.0开放授权认证下,系统资源能够被更多平台使用,通过安全API支持协议可以利于其他企业应用接入,方便相关数据供自身应用,提高了智慧校园系统的易用性,解决了资源分散重复存储的问题.

(Ⅳ)应用系统管理:通过高校与OAuth2.0权限控制体系相结合,对申请接入的应用分类,如果只需要简单资源,比如用户基本信息等资源的应用,可以由开发者全程自助完成应用申请,管理员事后审核;如果申请的资源包含敏感资源,开发者需要提交应用的主要功能和需要这些用户资源的场景,由管理员按实际情况允许或拒绝应用接入[3],增加了应用接入的灵活性,建设新的智慧校园认证应用管理模式,解决了智慧校园系统用户认证过程中的系统管理成本问题.

5 结论

我校智慧校园认证系统是基于OAuth2.0开放授权协议来实现第三方应用的接入和访问授权的,实现了标准的数据开放接口,有效解决了校园内各应用系统之间的信息孤岛问题,使得用户在使用第三方校园应用系统时,不再需要提供身份认证信息,OAuth2.0是我校智慧校园认证系统选择的保护用户的API和跨域的联合身份验证的方法[15],保证了用户认证信息的准确性、完整性和安全性,节约了系统开发成本和开发周期,实现了智慧校园应用系统的安全接入和用户信息的统一管理[13].经过流程测试,确定系统能够可控、安全地向第三方应用提供所需数据,达到了设计初期的目的,实现了校园信息化建设.

本文设计实现了一种基于OAuth2.0的高校统一认证系统,可以实现统一认证、双向单点登录,并集成开放平台账号登录,使外部开放平台账号能够接入认证.作为学校统一授权和细粒度业务集成的支撑平台,该系统的建设可以为用户提供更好的功能服务,即只需一个账号便可在各个系统中登录,在高校环境中,对用户的信息库建设也有帮助意义.后期继续改进两方面:一方面如何通过协议的优化、或与多种认证技术相结合的方法提高协议的安全性;另一方面如何利用支撑平台中用户各个维度的信息,分析用户喜好,从而为高校用户进行个性化推荐服务.该系统的设计实现对智慧校园的建设具有借鉴意义.

参考文献(References)

[1] 黄瑞钰.智慧校园建设方案与实现[D].华南理工大学,2014.

[2] 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 Huanxiao, GU Chunxiang, ZHENG Yonghui. Formal security analysis of Oath2. 0 authorization protocol [J]. Journal of Information Engineering University, 2014, 15(2): 141-147.
[3] 白雪松, 杜晋博, 王罡. 高校信息化环境下 OAuth 授权体系的研究与实践[C]. 北京:中国高等教育学会教育信息化分会第十二次学术年会论文集,2014.
[4] LEIBA B. OAuth web authorization protocol [J]. IEEE Internet Computing, 2012,16(1): 74-77.
[5] 刘姚. 基于 Spring 和 OAuth2.0 的第三方授权框架[J]. 计算机技术与发展, 2017, (3): 167-170.
LIU Yao. Investigation on third party authorization system based on Spring security andOAuth2. 0[J]. Computer Technology and Development, 2017, 27(3): 167-170.
[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 Chengkun, LIU Xiangdong, SHI Zhaojun. Optimization method for OAuth2. 0 protocol [J]. Netinfo Security, 2016, (9): 6-11.
[7] RAMA G M, KAK A. Some structural measures of API usability[J]. Software Practice and Experience, 2015, 45(1): 75-110.
[8] OpenAPI call description _ OAuth2. 0. [EB/OL]. http://wiki. open. qq. com/wiki/website/OpenAPI%E8%B0%83%E7%94%A8%E8%AF%B4%E6%98%8E_OAuth2. 0.
[9] CIRANI S, PICONE M, GONIZZI P, et al. IoT-OAS: An OAuth-based authorization service architecture for secure services in IoT scenarios[J]. IEEE Sensors Journal, 2015, 15(2): 1224-1234.
[10] OpenID Connect Core 1. 0 Incorporating Errata Set 1, Vol. 8, 2014. [EB/OL]. http://openid. net/specs/openid-connect-core-1_0. html.
[11] CHOI J, KIM J, LEE D K, et al. The OAuth2. 0 web authorization protocol for the internet addiction bioinformatics (IABio) database [J]. Genomics & Informatics, 2016, 14(1): 20-28.
[12] CHESS B, ARKIN B. Integrating user customization and authentication: The identity crisis [J]. IEEE Security & Privacy, 2012, 10(5): 82-85.
[13] 李熹, 王林. 数字校园统一认证平台的研究与应用[J]. 江西教育, 2016, 7-8.
[14] 张敏, 史纪强, 任恩茂, 宋建. OAuth2.0 在一体化管理平台资源共享中的应用研究[J]. 中国化工贸易, 2015: 182-183.
[15] LEIBA B. OAuth web authorization protocol [J]. IEEE Internet Computing, 2012, 16(1): 74-77.

(上接第 563 页)

[14] HUANG Chengqiang, MIN Geyong, WU Yulei, et al. Time series anomaly detection for trustworthy services in cloud computing systems[J/E]. IEEE Transactions on Big Data (Early Access), Boston, USA: IEEE, 2017. [2018-03-18] https://ieeexplore. ieee. org/document/7937930.
[15] LAPTEV N, AMIZADEH S, FLINT I. Generic and scalable framework for automated time-series anomaly detection[C]//International Conference on Knowledge Discovery and Data Mining. Sydney, Australia: ACM, 2015: 1939-1947.
[16] Twitter/AnomalyDetection. Code[EB/OL]. [2018-03-18]https://github. com/twitter/AnomalyDetection.
[17] Twitter/Breakout Detection. Code[EB/OL]. [2018-03-18]https://github. com/twitter/BreakoutDetection.
[18] Netflix/Surus. Code[EB/OL]. [2018-03-18]https://github. com/Netflix/Surus.
[19] CANDÈS E J, LI XD, MA Y, et al. Robust principal component analysis[J]. Journal of the ACM, 2011, 58(3): No. 11(1-37).
[20] Linkedin/luminol. Code [EB/OL]. [2018-03-18] https://github. com/linkedin/luminol.
[21] CHANDOLA V, CHEBOLI D, KUMAR V. Detecting anomalies in a time series database [R]. University of Minnesota, Technical Report 12, 2009.
[22] WELFORD B P. Note on a method for calculating corrected sums of squares and products [J]. Technometrics, 1962, 4(3): 419-420.
[23] Computing Systems Data. Yahoo, S5 - A Labeled Anomaly Detection Dataset, version 1. 0[EB/OL]. [2018-03-18] http://webscope. sandbox. yahoo. com/catalog. php? datatype=s&did=70, 2015.