图书馆行业 OpenAPI 利用的权限控制*

贾西兰 郭建峰
北京师范大学图书馆 北京 100875

〔摘要〕分析图书馆行业 OpenAPI 利用的类型、OpenAPI 来源、权限控制需求;介绍典型的 OpenAPI 访问权限控制途径:IP 地址范围控制、授权(OAuth、API key)、访问频次控制,分析其适用性,并阐述它们在图书馆行业的应用前景。结合北京师范大学图书馆在解决书目数据共享中的 OpenAPI 权限控制实践,给出基于 X-Services 的按需授权控制实例。
〔关键词〕图书馆行业 OpenAPI 权限控制 安全
〔分类号〕G250.71

Access Control of OpenAPI in Library Field

Jia Xilan Guo Jianfeng
Library of Beijing Normal University, Beijing 100875

〔**Abstract**〕The types of OpenAPI application in library field are analyzed, as well as OpenAPI providers and access control requirements. The main ways of OpenAPI access control are described: by IP address, authorization(OAuth, API key) and access times control. Their applicability and prospect in library field are discussed. With the OpenAPI access control practices of sharing catalog resources in Beijing Normal University Library, an example of X-Services authorization on demand is given.
〔**Keywords**〕library field OpenAPI access control security

OpenAPI 是在互联网上开放的应用编程接口,在 Web 2.0 环境下得到了快速发展。OpenAPI 基于 HTTP 协议,以 XML、JSON 等格式返回信息,为在 Web 上构建新的应用、达成各异构系统间的资源共享和互操作提供了方便。OpenAPI 正在成为图书馆行业独立系统间互通互联的新选择。然而,随着 OpenAPI 数量的激增和种类的丰富,开放背后的安全问题也显现出来,有些 OpenAPI 可以对整个互联网开放,任何用户都可以访问,更多的 OpenAPI 涉及受保护资源或关键性业务环节,只有被授权的用户才能访问。如何控制 OpenAPI 利用的权限,避免非法访问、越权访问、信息泄密、数据篡改、无度滥用等问题出现,是亟待解决的问题。笔者通过 CNKI、Google Scholar、SpringerLink 等进行了文献调研,没有发现系统论述 OpenAPI 利用权限控制方面的文章。相关研究有 OAuth 认证[1]、Web 2.0安全问题[2]、云安全[3]等。本文试对图书馆行业 OpenAPI 利用的权限控制问题进行探讨。

1 图书馆行业 OpenAPI 的使用分析

1.1 图书馆行业 OpenAPI 的常见应用类型分析

OpenAPI 在解决“信息孤岛”问题、方便异构系统互通互联上的优势与数字图书馆建设中的瓶颈问题形成了高度吻合的针对性解决途径。

目前,图书馆行业 OpenAPI 应用的常见类型主要有如下 6 种:①整合检索:将多个资源系统的数据通过统一的检索入口提供一站式检索服务,如:资源门户、联合书目等。②信息导航:将多个系统中的信息按照某种秩序逐层组织起来,如:按类别、按顺序等,引导用户通过层层深入的方法获得感兴趣的信息。③资源集成:将分散存在于多个系统中的数字资源作为一个有机的整体,通过一定的关联条件将它们无缝地集成在一个应用系统中使用。如:在 OPAC 上集成豆瓣网的书评。④延伸传统图书馆功能:利用多种网络协议,将传统图书馆的服务与功能延伸到更多的平台和设备上。如基于 WAP 的手机移动图书馆[4]、短信服务平台

*本文系《图书情报工作》杂志社出版基金项目“数字图书馆 OpenAPI 的标准化和权限控制技术”(项目编号:2011CB004)研究成果之一。
收稿日期:2011-10-11 修回日期:2011-12-25 本文起止页码:21-25 本文责任编辑:徐 健

等。⑤连接自助设备:我们通常把代替图书馆工作人员为读者提供服务的机器称为自助设备,这些机器由读者自行操作,如自助借还、自助办证等[5]。⑥参与其他业务流程:以多个独立系统为基础,构建无缝的业务流程,如:学校建立了毕业生离校一条龙网上办理系统,图书馆这一环节的退证手续通过 OpenAPI 融入到这个流程中。

分析这些应用类型,前三种需要的 OpenAPI 以具有搜索功能的一类为主,后三种涉及业务流程、数据更新、私人信息查询等。在一般情况下,搜索性 OpenAPI 无需写数据,应用时数据被破坏的风险较小,相对安全,可以有比较广泛的开放范围,控制他们的访问权限主要是考虑资源的合法使用对象是什么,避免版权、使用权等的越权纠纷。对于后三种应用类型使用的 OpenAPI,不仅要考虑合法的使用对象,还要考虑数据安全,考虑第三方使用不当会带来的风险,故而应该有比较严格的开放范围和条件。

1.2 不同提供者 OpenAPI 的权限控制分析

不同的 OpenAPI 提供者,对于权限控制的需求有所不同,下面简单进行分析。

- 互联网上的公共网络服务提供者:他们在整个互联网范围提供 OpenAPI 服务,如:Google、亚马逊、豆瓣网等。他们将资源分为公共资源和受保护资源。对于公共资源,可以通过 OpenAPI 直接访问,或者完成一个应用程序注册,取得访问许可。而对受保护资源,如托管资源,要得到资源拥有者的授权。

- 联合体或局部公共服务体系:他们是由多个实体机构共建的资源联盟或是由权威机构主管建设的服务体系,在本联合体或公共服务体系的范围内提供 OpenAPI 服务。如:作为中国高等教育文献保障系统,CALIS 三期将建设 CALIS 数字图书馆云服务平台[6],计划将各项服务封装为统一的 OpenAPI 对外服务,也将采取托管服务的方式统一规范来自各成员馆和其他独立服务提供商提供的 OpenAPI,以方便各个成员馆对其调用。

- 数据库提供商:他们是商业化运营的数字资源服务公司,图书馆从那里购买有使用范围限制的电子书刊、数据库、多媒体资源库等,如:CNKI、读秀。他们提供的 OpenAPI 一般与资源的使用范围相同,典型的是按照 IP 地址授权使用,如:在校园网范围内。有些数据库提供商会将搜索类 OpenAPI 的使用放权到互联网范围,如:CNKI 的 KDE [7]提供的 OpenAPI 可以任意调用,起到揭示资源的作用,在打开全文时才限制使用范围;而读秀的搜索类 OpenAPI 就不接受资源使用范围之外的 IP 地址发起的调用请求。

- 图书馆:图书馆对自身可管理的系统添加开放接口,提供的 OpenAPI 服务分为两种:①图书馆使用的商业软件系统本身带有 OpenAPI 服务功能,基于已有 OpenAPI 权限控制体系,图书馆自己决定是否开放,如何开放;②图书馆基于商业软件系统或自建系统自己开发的 OpenAPI,如:厦门大学图书馆[8]。对于自己开发的 OpenAPI,图书馆可以设计合适的权限控制体系。

在对 OpenAPI 的访问进行权限控制时,以什么为单位进行控制是个基础的问题,常见的形式有:按照 OpenAPI 具有的功能划分的分层控制、按照 OpenAPI 涉及的资源划分的分类控制、将每个 OpenAPI 单独控制的独立控制、将所有 OpenAPI 集中控制的整体控制等。

2 典型的 OpenAPI 访问权限控制分析

由于 OpenAPI 不是由终端用户在浏览器地址栏中直接通过网址访问,而是由网页、客户端脚本、服务器端脚本等应用程序调用的,在其访问权限的控制上与终端用户的情况有些不同,目前尚无成熟的标准和方案。

2.1 通过 IP 地址或域名控制 OpenAPI 访问权限

首先定义一个允许访问的 IP 地址或域名范围,当发起 OpenAPI 访问请求的服务器或客户端的 IP 地址或域名属于这个范围时,访问得以顺利进行,反之会被拒绝。这种权限控制建立在对 IP 地址或域名的充分信任基础上,适用于使用者有固定 IP 或域名的情况。它在技术实现上相对容易,可以通过 Web 服务器配置实现,Apache、Microsoft IIS 等常用 Web 服务器都具有这样的功能,但作为网站级控制,它不易实现精确控制;另一种途径是通过应用系统自身控制,它可以将不同 OpenAPI 分别控制。高校图书馆提供的 OpenAPI 使用 IP 限制的比较多,比如英国剑桥大学图书馆的很多 OpenAPI 都限制 IP 范围内使用。

2.2 通过授权控制 OpenAPI 访问权限

授权是指给使用 OpenAPI 的一方赋予一定的许可权,是以使用者为对象进行控制的。但因为现实中有多种多样的控制需求,如何赋权,又如何进行验证、判断,是个比较复杂的问题。以下分析两种使用比较普遍的 OpenAPI 访问授权控制。

- OAuth OAuth (open authorization)[9]是一个以

资源为中心的通过 OpenAPI 访问受保护资源的安全开放授权协议。它很好地解决了这样一个问题：当一个资源中心存在多种托管资源时，谁能访问这些资源，不由 OpenAPI 服务提供者决定，必须由资源拥有者授权，而且可以对其中的部分资源单独授权，且需要收权和放权都方便。一般情况下，资源拥有者访问他们的资源是通过“用户名/密码”方式的，如果授权时把“用户名/密码”提供给资源使用者将非常不安全，等于把对资源的全部使用权都开放了，而且若想收回授权，只有修改密码才可以。OAuth 可以不通过“用户名/密码”解决这个问题。

OAuth Core 1.0 发布于2007年12月4日[10]，最新版的 The OAuth 2.0 Authorization Protocol 发布于2011年7月25日[11]，它定义了4种角色：①资源拥有者：可以为受保护资源批准访问权限的实体；②资源服务器：承载被托管的受保护资源，并且能够接受和响应使用访问令牌（access tokens）访问受保护资源的请求；③客户端：持有资源拥有者的授权、代表资源拥有者发出请求访问受保护资源的应用程序；④授权服务器：在成功地验证了资源拥有者并获得授权后，将访问令牌发放给客户端。

OAuth 的开放授权过程如图1所示：

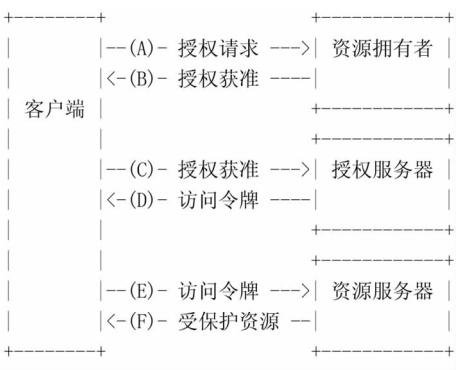

图1 OAuth 开放授权过程

（A）客户端向资源拥有者请求授权；（B）客户端接受一个资源拥有者的授权获准；（C）客户端用授权获准向授权服务器请求访问令牌，并接受身份验证；（D）授权服务器验证客户端的合法性和授权获准的有效性后向客户端发放访问令牌；（E）客户端用访问令牌向资源服务器请求访问受保护资源；（F）资源服务器验证了访问令牌的有效性后提供所请求的服务。访问令牌的生命周期是短暂的，在资源拥有者授权的允许访问时间段内，要依靠刷新令牌不断地更新，同时它包含了授权访问的资源范围。

OAuth 协议很快得到了工业化的应用，首先是一些大型互联网服务提供商宣布对 OAuth 的支持：Google、Twitter、亚马逊、豆瓣网，等等。OAuth 对于图书馆行业的意义也很明显。图书馆有很多受保护资源，如：机构知识库、学位论文、课件、个人图书馆、学科馆员专题资源等，目前对这些资源的使用控制大多采用比较简单的方式，有些资源拥有者不愿意将资源提供给图书馆，典型的例子是课件，很多课件要求仅提供给与课程相关的师生使用，目前还难以做到。OAuth 将在图书馆加大服务深度、优化个性化服务上起到关键的推进作用。

- API key API key 是 OpenAPI 服务提供者为 OpenAPI 服务使用者发放的授权许可。API key 一经申请获得，相对稳定，不会自动失效。API key 一般由一个字母与数字混合的字符串组成，使用上比较简单，当程序请求 OpenAPI 服务时，把相应的 API key 作为参数传递过去即可。支持 API key 授权方式的互联网服务提供者更为普遍，但目前并没有什么标准形成。由于 API key 不是动态变化的，一经泄露就会产生伪装用户，进行非法使用，所以视不同需求常与 https 加密传输、签名验证等结合使用；有些网站也将 API key 与申请使用 API key 的网站域名绑定在一起进行验证。API key 适用于 OpenAPI 使用者比较固定、OpenAPI 服务提供者能够进行使用授权的情况。

2.3 通过频次限制控制 OpenAPI 访问权限

频次控制是在上述控制的基础上对单位时间内的访问频次规定上限，超出这个上限时采取禁用、停用等处理。比如 LibraryThing API 限制用户访问次数每天不超过1 000次，OCLC 的相关 API 非商业用户访问次数每天不能超过500次，通过 IP 认证的用户每天不能超过1 000次。每天超出正常范围的 OpenAPI 访问量最明显地带来两个问题：①对服务器承受能力的冲击；②非法数据套录。因此，采用频次控制在一定程度上可以防止资源被滥用或恶意使用。

2.4 OpenAPI 访问权限控制实例

在实际应用中，以上几种控制 OpenAPI 访问权限的方法是可以组合使用的，下面分析两个具体例子。

Google Books API Family（Labs）[12]：Google 支持 OAuth，原来的 AuthSub 和 ClientLogin 已经作为旧协议，不再提倡使用。Google 新推出的“Books API v1（Labs）”有如下规定：首先，使用者要向 Google 注册登记相关的应用系统，应用系统的域名是将来访问 OpenAPI 的检验条件之一。Google 提供 OAuth 和 API

key 两种方式识别使用者的应用系统,具体细节是:如果应用系统发出的请求涉及受保护资源,则必须提供一个 OAuth 访问令牌,是否还提供 API key 不是必要条件;如果应用系统发出的请求不涉及受保护资源,则可以依照最方便的方式选择下面三种方法之一:提供一个 OAuth 访问令牌、提供一个 API key、两者都提供。另有嵌入式图书预览 API(embedded viewer API)、动态链接 API 对互联网自由开放,不需要任何验证。由此可见:Google 比较好地兼顾了公共资源与托管资源的开放区别。

豆瓣[13]:豆瓣网要求每个 API 的使用者都申请一个 API key,在 API 请求中用 apikey 参数传递 API key,同时控制请求频次,每分钟请求不超过 40 次,需要超过者另外申请。不申请 API key 也可以调用 API,但被限制为每分钟请求不超过 10 次。当第三方应用需要通过 API 访问或修改受保护的用户数据时,需要通过 OAuth 认证机制来获得用户的授权。由于在豆瓣的 OAuth 访问令牌中包含了 API key 参数,所以用 OAuth 访问令牌时不需要再进行 API key 认证。

每种权限控制方式都会消耗一定的计算资源和网络资源,图书馆自提供 OpenAPI 服务采取什么控制方式,要根据安全要求、实施难度、平均响应速度、硬件条件等综合考虑。

3 OpenAPI 权限控制实践

随着 OpenAPI 应用的普及,OpenAPI 权限控制的需求呈现多样化,需要根据不同的系统环境在实践中探索可行的解决途径。本实践根据建立全国师范院校图书馆资源共享联盟的需求进行:按照计划,联盟中院校的书刊信息要共知、共享(包括联合检索),需要图书馆集成管理系统开放接口。北京师范大学(以下简称"北师大")图书馆使用的图书馆集成管理系统是 ExLibris 公司的 ALEPH 500,该系统有一组称为 X-Services 的 OpenAPI 接口,提供了书目检索、读者认证、预约、续借、数据更新等多项服务,北师大图书馆利用这组接口进行了多项应用拓展和馆内系统间互联,包括整合检索、资源集成、手机图书馆等。ALEPH 500 系统本身提供了通过 IP 地址和用户授权设置对 X-Services 访问权限进行整体控制的机制。但是,如果对众多的联盟成员进行 IP 地址和用户授权的设置与后期更新维护,将带来非常大的管理工作量,而且影响系统运行速度。

笔者借鉴了通过 API key 控制 OpenAPI 访问权限的思想,设计并实现了通过 API key 控制 X-Services 访问权限的方案,不通过 IP 地址过滤,将 X-Services 的服务按需增加授权、开放。原理描述如下:

- 建立 API key 数据库表 APIkeyop,记录每个 API key 允许访问哪些 X-Services 服务操作。
- 建立服务器端 API key 验证功能,接受并验证 X-Services 服务请求。
- 将 X-Services 请求的 URL 进行变换,首先指向验证程序 aleph_services,同时附加 API key 参数。如检索题名中含有 java 一词的图书,原始 X-Services 请求表达是:http://server:port/X?op=find&code=wti&request=java&base=abc01(op——操作,code——字段,request——检索词,base——数据库)。变换后为:http://server:port/services/aleph_services?op=find&code=wti&request=java&base=abc01&id=2abcde873438475(id——API key)。
- 记录 API key 访问 log,用于频次控制。

关键代码(Perl 语言):

建立数据库表 APIkeyop($dbh——数据库句柄,ido——API key + op 参数,auth——授权(Y/N))

```
$dbh->do("CREATE TABLE APIkeyop(ido text,auth text,ctime)");
$dbh->do("CREATE UNIQUE index APIkeyop_id on APIkeyop(ido)");
```

获取传递的参数

```
$cgi=new CGI;
my $id=$cgi->param("id");
my $op=$cgi->param("op");
my $id1=$id.$op;
```

从 APIkeyop 中查找 API key 对 op 操作的授权

```
$sqr=$dbh->prepare("SELECT ido,auth FROM APIkey op where ido='$id1'");
$sqr->execute() or die $dbh->errstr;
```

如果授权为"Y",构建访问 X-service 的 URL,并发送请求:

```
while(@t=$sqr->fetchrow_array){
if(@t[0] eq $id1){
$auth1=@t[1];}
}
if($auth1 eq "Y"){
my @params=$cgi->param;
my $url="$bnuopac_base/X?";
```

```
foreach my $pa (@params) {
if($pa ne "id"){
$url .= '&'.$pa.'='.$cgi->param($pa);}
}
$content = get $url;
die "Couldn't get $url" unless defined $content;
}
返回结果
print $content;
```

效果验证表明:持有 API key 的联盟成员,通过验证程序 aleph_services 请求 X-Services 服务时,可以顺利使用已经授权的 X-Services 服务操作,当越权请求其他 X-Services 服务操作时,将获得"no Authentication"提示,同时,频次控制也有效控制了过度频繁的异常访问。本方案可以扩展到对任意个体和组织的 X-Services 服务访问授权,不同的 API key 可以形成不同的 X-Services 服务权限组,满足不同应用需求。而且可以将授权精确到单次服务请求操作,如:检索服务有两个请求操作:①检索,返回命中数;②在命中结果中获取书目详细信息,本方案可以只授权前者操作。本方案也可以借鉴到其他应用系统对 OpenAPI 权限控制的实践中。

4 结 语

OpenAPI 在图书馆行业有极好的应用前景,通过 OpenAPI 实现独立系统间的互通互联可以产生单一系统无法达到的增值性应用,将推动图书馆服务的创新、资源的充分利用以及对外合作的共赢。OpenAPI 应用的权限控制直接影响着 OpenAPI 的深入应用,目前,全球范围内相关的工业化标准正在发展和形成之中,图书馆行业也应当积极参与其中,努力实践,不仅要利用已有的 OpenAPI,也要开发自己的 OpenAPI、改进现用的 OpenAPI,解决发展中的实际问题,发挥本行业的信息服务优势,推动服务模式的创新以及相关标准的建立和应用。

参考文献:

[1] 张卫全,胡志远. 浅析作用于 Web2.0 安全防范的 OpenID 和 OAuth 机制[J]. 通信管理与技术, 2011(2):15-18.

[2] 徐毅,王红阳. Web2.0 安全性刍议[J]. 软件世界,2006(19):62-63.

[3] Cloud identity summit[EB/OL]. [2011-08-01]. http://www.cloudidentitysummit.com/ 2011-07-18.

[4] 林颖,孙魁明. 基于 WAP 的图书馆移动信息服务体系及 WAPOPAC 应用实例[J]. 现代图书情报技术, 2007(9):80-83.

[5] 张红,只莹莹. 利用自助模式提升读者服务[J]. 数字图书馆论坛,2010(12):51-55.

[6] 王文清,陈凌. CALIS 数字图书馆云服务平台模型[J]. 大学图书馆学报,2009(4):13-18.

[7] KDE V2.0 接口使用帮助[EB/OL]. [2011-08-01]. http://kde.cnki.net/KDEService/Search/Help/.

[8] 肖铮,陈晓亮. 厦门大学图书馆馆藏书目信息 API 开发实例及其应用[EB/OL]. [2011-08-01]. http://dspace.xmu.edu.cn/dspace/bitstream/2288/7073/2/%E5%8E%A6%E9%97%A8%E5%A4%A7%E5%AD%A6%E5%9B%BE%E4%B9%A6%E9%A6%86%E9%A6%86%E8%97%8F%E4%B9%A6%E7%9B%AE%E4%BF%A1%E6%81%AFAPI%E5%BC%80%E5%8F%91%E5%AE%9E%E4%BE%8B%E5%8F%8A%E5%85%B6%E5%BA%94%E7%94%A8.pdf.

[9] OAuth[EB/OL]. [2011-08-01]. http://en.wikipedia.org/wiki/OAuth.

[10] The OAuth 1.0 Guide[EB/OL]. [2011-08-01]. http://hueniverse.com/oauth/guide/.

[11] The OAuth 2.0 Authorization Protocol draft-ietf-oauth-v2-20[EB/OL]. [2011-08-01]. http://tools.ietf.org/html/draft-ietf-oauth-v2-20.

[12] Google Books API Family (Labs)[EB/OL]. [2011-08-01]. http://code.google.com/intl/zh-CN/apis/ books/docs/getting-started.html.

[13] 豆瓣 API[EB/OL]. [2011-08-01]. http://www.douban.com/service/apidoc/.

〔作者简介〕贾西兰,女,1957 年生,研究馆员,发表论文 10 余篇。
郭建峰,男,1970 年生,副研究馆员,发表论文近 10 篇。