第 19 卷 第 19 期 2019 年 7 月
1671—1815(2019)019-0196-05

科 学 技 术 与 工 程
Science Technology and Engineering

Vol. 19 No. 19 Jul. 2019
© 2019 Sci. Tech. Engrg.

引用格式:姜建武,胡 垚,李景文. 基于 Request Body 的 Open API 安全认证机制[J]. 科学技术与工程,2019,19(19):196-200
Jiang Jianwu, Hu Yao, Li Jingwen. Open api security authentication mechanism based on requestbody[J]. Science Technology and Engineering, 2019, 19(19): 196-200

基于 Request Body 的 Open API 安全认证机制

姜建武 胡 垚 李景文*
(桂林理工大学测绘地理信息学院,桂林 541004)

摘 要 为解决当前 Open API 面临的身份伪造钓鱼攻击、账户与业务信息泄露和 API 平台恶意攻击等问题,提出了一种基于 Request Body(请求体)的 API 安全认证机制。该机制由双重签名验证、请求体加密、URI 验证、接口权限认证和异常侦测五部分组成,用于防范钓鱼网站诱骗、加固用户数据传输安全和提升 API 平台抵御攻击的能力。通过在线测试和实际项目验证,表明该机制能够在保证 API 认证速度的同时保证用户和接口的安全。

关键词 Open API 安全认证 RestFul API 信息加密
中图法分类号 TP393; **文献标志码** A

开放的互联网、更加广泛的数据共享需求和构建更加完善的产品生态的要求共同推动了 Open API 技术的发展[1]。Open API 技术的兴起,为搭建 Mashup 风格的 Web 应用提供了更大的选择空间[2],同时,OpenAPI 也是平台发展、资源共享的重要途径[3]。但是,随着基于 Open API 的应用大量涌现,其安全隐患也同样暴露出[4],由于 Open API 开放的特性导致的账户信息泄露[5]、身份伪造钓鱼攻击[6]、业务信息泄露[7]和平台恶意攻击等问题已经大大威胁到了 Open API 的安全。刘武等针对安全认证存在开放重定向攻击漏洞问题,设计了一种新的认证机制,提高了认证的安全性[4]。雷蕾等、刘俊提出在 Open API 中采用非对称加密和令牌机制保证 API 的安全性[8,9]。刘树凯等针对当前 JavaScript API 保护机制缺失的问题,提出了一种轻量级的 PMJA 保护机制,能够防止 JS API 被恶意利用[10]。王玉等通过 https 和用户 Token 双重验证实现 API 的安全防护[11]。以上研究为保证 Open API 的安全提供了有效的措施。本文在大量工程实践的基础上,从 URI 验证和参数传递时就对 API 进行防护加固的角度,提出了一种基于 Request Body(请求体)的 Open API 安全认证机制。该机制通过双重签名验证、请求体加密、出/入 URI 验证、接口权限认证和异常侦测锁定五个方面保证 API 服务器和使用者的安全。其中双重签名认证和请求体加密用于防止账户信息和业务信息泄露和遭篡改,出/入 URI 验证用于防止用户遭受身份伪造钓鱼攻击,接口权限认证和异常侦测锁定用于降低平台被恶意攻击的风险。

2019 年 1 月 10 日收到 国家自然科学基金(41461085)、桂林市科学技术局项目(20170220)和广西测绘局项目(2018-B-02)资助
第一作者简介:姜建武(1992—),男,汉族,安徽阜阳人,硕士,助教。E-mail: fengbuxi@glut.edu.cn。
* 通信作者简介:李景文(1970—),男,汉族,内蒙古赤峰人,博士,教授。E-mail: lijw@glut.edu.cn。

1 基于 Request Body 的安全认证流程

1.1 Request Body 的构成

引入请求体的概念是为了保证 API 入口的唯一性和参数的安全封装。由于 Request Body 采用了非对称加密和多次加密封装,因此可以最大程度的防护用户的信息和业务信息被篡改和截获,同时每个 Request Body 的有效期都有时间限制,避免了恶意用户的重复攻击。Request Body 包含接口参数、业务参数和账户参数三个部分,其结构如图 1 所示。

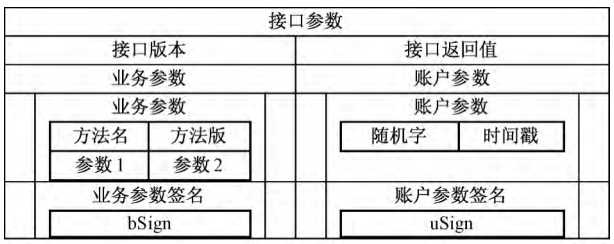

图 1 Requuest Body 结构
Fig. 1 Requuest Body structure

1.1.1 接口参数

接口版本信息定义了接口的版本号和接口的返回值类型,接口的版本号不同于业务参数中的方法版本号,接口版本号作用于整个接口的级别,方法版

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

本号限定于具体的业务方法。接口返回值分为 json 和 xml 两种，默认的返回值类型为 json，用户可根据需要自定义为 xml 类型。

1.1.2　业务参数

业务参数由两部分组成，第一部分包括了该业务对应的方法名称、方法的版本和具体的业务参数，第二部分是根据第一部分参数、时间戳、随机字符串、公匙和用户 ID 经组合并采用 sha1 算法生成的业务参数签名。业务参数签名能够验证请求参数是否是对应用户在某一时间发起的业务请求。

1.1.3　账户参数

账户参数同样由两部分组成，第一部分包含用户的 OpenID、时间戳和随机字符串，第二部分是将以上参数顺序组合并加入公匙后经 sha1 算法生成的签名，该签名的作用是验证用户的合法性。

1.2　基于 Request Body 的安全认证流程

Request Body 的安全认证由请求体加密、出/入 URI 验证(返回 UrlCheck 字段)、双重签名验证、接口权限认证和异常侦测锁定五部分组成，具体流程如图 2 所示。

用户首先根据 Request Body 的参数生成规则生成 Request Body，然后请求 API，Request Body 的构建方法见 1.3 节。异常侦测引擎和日志记录引擎在整个 API 进程中都处于激活状态，用于收集、统计、记录 API 异常信息。

1.2.1　请求体加密

对请求体加密是为了防止请求体被非法劫持并遭恶意篡改，对 API 平台的稳定性和用户权益产生不利影响。请求体的加密采用非对称加密算法 RSA，加密公匙和私匙对应用户公私和私匙[12]，用户的公私存放于数据库，私匙存放于服务器，采用公、私匙分开存储能够避免当数据库被攻破后用户 API 认证信息的泄露。

1.2.2　出/入 URI 验证

通过授权 API 的服务域名和用户发起域名防止 API 被其他人恶意使用。此处的出 URI 是指系统授权的用户方域名，即用户发起的域名(或 IP)，入 URI 是指 API 服务所在的服务器的域名(或 IP)。对于正确的请求，系统将在返回值中返回 UrlCheck 字段，用户可根据 UrlCheck 字段判断访问的 API URL 是否是合法的 URI。UrlCheck 字段根据用户的公、私匙计算，恶意服务器无法获取私匙，因此可以避免用户遭受身份伪造钓鱼攻击。

1.2.3　双重签名验证

双重签名即业务参数签名和账户参数签名，验证的顺序是先验证账户参数签名，其次验证业务参数签名。账户参数签名验证不通过则终止操作，返回错误信息并添加异常记录[13]。业务参数签名主要的目的是防止请求体被破解后业务参数遭到非法篡改。通过双重签名验证，保证了账户和业务参数

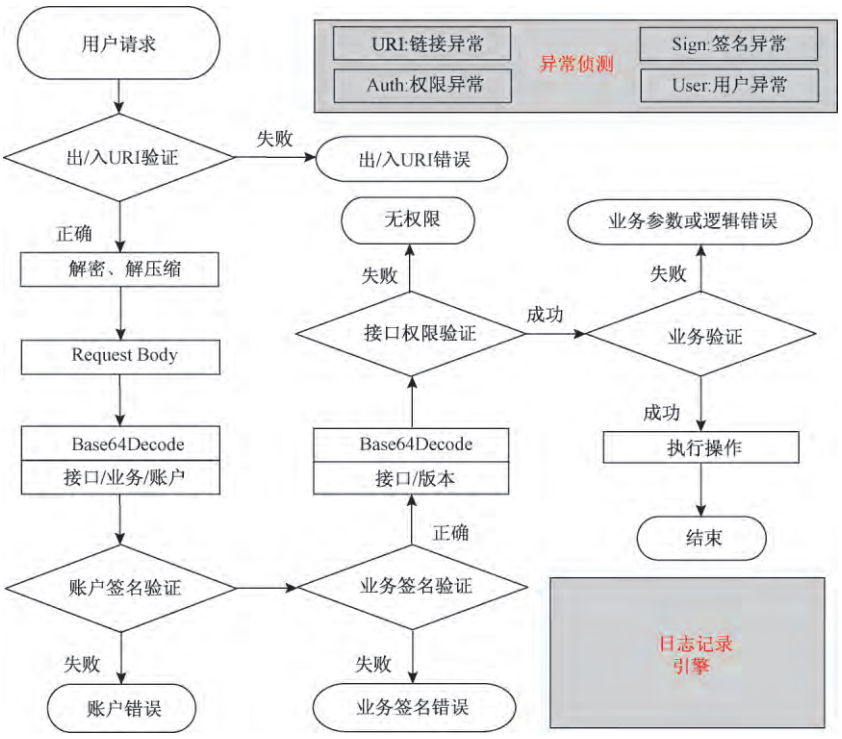

图 2　Request Body 的安全认证流程

Fig. 2　Security certification process of Request Body

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

的安全[14]，即使被截获，也无法解密信息。

1.2.4 接口权限认证

接口权限验证可以屏蔽用户的非法请求，并可侦测该用户的请求是否正常，如果用户频繁发起无权限接口的请求，则可初步判断该用户接口行为异常，调用 Auth 异常锁定用户账户。

1.2.5 异常侦测锁定

异常侦测锁定活跃于整个 API 进程，系统设定 5 min 内同一异常连续出现 10 次，则判定为异常成立，系统将对用户账户进行锁定。目前异常分为 Uri（出/入 URI 异常）、Sign（签名异常）、Auth（接口权限异常）、User（用户账户异常）和 Business（业务参数异常）5 种。结合接口权限和异常侦测锁定功能可以降低平台被恶意攻击的风险。

1.3 基于 Request Body 的安全认证实现

1.3.1 用户构建 Request Body

用户在构建 Request Body 时需要用到以下参数，相关参数及说明如表 1 所示。

表 1 用户构建 Request Body 时的相关参数说明

Table 1 Description of related parameters when the user constructs the Request Body

参数名	参数类型	意义	说明
OpenID	String	开放平台 ID	由 Open API 平台分配
NoStr	String	随机字符串	字符串长度为 10～30
TimeStamp	Int	时间戳	互联网时间戳
PublicKey	String	用户公匙	由 Open API 平台分配

用户构建 Request Body 分为构建接口参数、构建账户参数、构建业务参数、构建请求体、请求体加密 5 个步骤组成。

（1）构建接口参数

①构建接口参数 Json 字符串 InterfaceJson。

```
{
    "Version": "接口版本",
    "ReturnType": "接口返回类型 json/xml"
}
```

②构建接口参数 Interface。

```
string Interface = SCompress ( Base64Encode( InterfaceJson) )
```

其中 Base64Encode 是一种字符串编码方式，SCompress 是本文开发的字符串压缩方式，主要目的是减少传输时字符串的长度。

（2）构建账户参数

①构建账户参数字符串 AccountStr。

```
string AccountStr = OpenID + NoStr + TimeStamp
```

②获取账户参数签名 aSign。

```
string aSign = StrToLower( Sha1 ( Md5 ( AccountStr) + PublicKey) )
```

其中 StrToLower 函数是将字符串转换为小写，Sha1 和 Md5 是不可逆加密函数。

③构造账户参数 Json 字符串 AccountJson。

```
{
    "NoStr": "随机字符串",
    "TimeStamp": "时间戳",
    "aSign": "账户参数签名"
}
```

④获取账户参数 Account。

```
string Account = SCompress( Base64Encode( AccountJson) )
```

其中 Base64Encode 是一种字符串编码方式，SCompress 是本文开发的字符串压缩方式，主要目的是减少传输时字符串的长度。

（3）构建业务参数

①构建业务参数 Json 字符串 BusinessJson。

```
{
    "Function": "方法名",
    "FunVersion": "方法版本",
    "Param1": "参数 1",
    "Param2": "参数 2",
    "Param…": "参数 n"
}
```

②获取业务参数压缩字符串 Business。

```
string Business = SCompress( Base64Encode( BusinessJson) )
```

③获取业务参数签名 bSign。

```
string bSign = StrToLower( Sha1( Md5( Business + OpenID + NoStr + TimeStamp) + PublicKey) )
```

（4）构建请求体

①组合接口参数、账户参数和业务参数，获取请求体 Json 字符串 RequestBodyJson。

```
{
    "Interface": "接口参数 Interface",
    "Account": "账户参数 Account",
    "Business": "业务参数 Business",
    "bSign": "业务参数签名 bSign"
}
```

②获取压缩字符串 RequestBodyBase64。

```
string RequestBodyBase64 = SCompress( Base64Encode( RequestBodyJson) )
```

（5）请求体加密

采用 RSA 算法对请求体加密，其中 RSA 公、私匙对应 Open API 平台分配给用户的公、私匙。

获取请求体的加密字符串 RequestBody。

```
String RequestBody = Rsa ( RequestBody, PublicKey)
```

1.3.2 系统验证 Request Body

系统验证即构建 Request Body 的逆过程，在此

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

基础上增添了 URI 验证、异常锁定和认证缓存功能，本节主要讲述新增功能。

(1) URI 验证

用户发起 API 访问后，由服务器端获取目标 URI 和来源 URI，并与数据库进行比较，若不匹配则返回 URI 错误，并记录到异常侦测引擎中，当错误次数达到限额之后锁定该账户。为了方便用户判断 API 请求接口是否为合法 URI，防止钓鱼网站获取用户信息，系统将在每次正确请求的返回值中添加 UrlCheck 字段，其计算方法如下：

string UrlCheck = Rsa(OpenID, PrivateKey)

用户判断时，以 PublicKey 进行解码。

(2) 异常锁定

异常锁定机制是为了防止非法用户对 Open API 系统的恶意攻击，采用异常锁定机制可以提高攻击的难度，提升系统的安全和稳定。异常锁定机制的流程如图 3 所示。

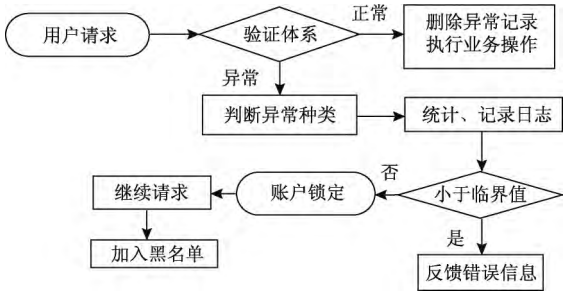

图 3　异常锁定机制流程图

Fig. 3　Abnormal locking mechanism flow chart

当用户的请求不通过时，系统根据异常的类型和时间进行分别统计，当某一类型达到临界值时将对用户账户进行锁定，处于锁定状态的用户如果继续发起请求达到 30 次，系统将对发起请求的 IP 做黑名单处理，从服务器层面拒绝该 IP 的请求。当用户异常在未达到临界值之前出现正确的访问，系统将清空该用户已保存的异常统计信息[15]。

(3) 认证缓存

为了提升 Request Body 认证的效率和 Open API 的访问速度，采用 Redis 技术对验证通过的用户授权做缓存处理，缓存时间为 7 200 s。缓存授权只保存用户账户和接口授权，业务参数授权不做缓存，用户每次请求都将验证业务签名。

2　Request Body 的安全认证效率实验

基于 Request Body 的安全认证机制已经在酷我游开放平台稳定运行 1 年，同时作者对认证的效率进行了实验。实验过程如下。

开发自动化测试脚本，测试用户 10 个，每个账户发起请求 100 次，每个账户 API 检验用时的平均值如图 4 所示。

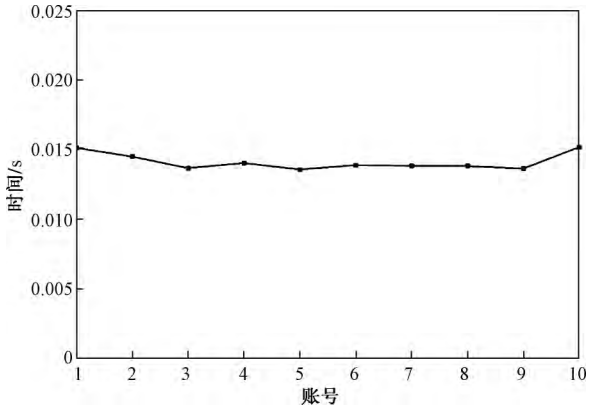

图 4　RequestBody 认证用时

Fig. 4　RequestBody authentication time

实验表明，API 平均验证时间为 0.014 11 s，能够在保证 API 安全认证速度的同时，满足 API 安全的需要。

3　结论

围绕保证 Open API 的安全，研究了基于 Request Body 的安全认证机制，该机制主要通过双重签名验证、请求体加密、出/入 URI 验证、接口权限认证和异常侦测锁定 5 个方面保证 API 服务器和使用的安全，通过实际验证得出如下结论。

(1) 基于 Request Body 的安全认证机制的 UriCheck 对防范钓鱼网站具有很大的作用。

(2) 双重签名验证机制能够极大地提升用户权限安全和业务安全，防止信息泄露。

(3) 异常锁定和接口权限认证机制能够极大地提升 API 服务器抵御恶意攻击的能力。

(4) 请求体加密可以防止系统参数遭劫持。

(5) 基于 Request Body 的安全认证机制能够在保证 API 速度的同时，满足 API 安全的需要。

参　考　文　献

1　欧阳璟. 把握互联网的下一次趋势——“中国互联网的 OpenAPI”研讨会选录[J]. 程序员，2008(7)：42-44

Ouyang Yu. Grasping the next trend of the internet— “opening the internet of China” seminar selection [J]. Programmer, 2008 (7): 42-44

2　韩　冷. 面向 OpenAPI 的开放能力管理平台[D]. 天津：天津大学，2010

Han Leng. Open capability management platform for OpenAPI [D]. Tianjin: Tianjin University, 2010

3　李　斌. 面向室内地图信息服务的 OpenAPI 的设计和实现[D]. 北京：北京邮电大学，2013

Li Bin. Design and implementation of OpenAPI for indoor map infor-

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

mation service [D]. Beijing: Beijing University of Posts and Telecommunications, 2013

4 刘 武, 王永科, 孙东红, 等. 开源智能终端认证漏洞挖掘及登录认证改进 [J]. 清华大学学报(自然科学版), 2017, 57(9): 897-902
Liu Wu, Wang Yongke, Sun Donghong, et al. Open source intelligent terminal authentication vulnerability mining and login authentication improvement [J]. Journal of Tsinghua University(Science and Technology), 2017, 57(9): 897-902

5 Mariantonietta L P, Febio M, Daniele S. A survey on security for mobile devices [J]. IEEE Communications Surveys & Tutorials, 2013, 15(1): 446-471

6 William E, Machigar O, Patrick M. Understanding Android security [J]. IEEE Security and Privacy, 2009, 7(1): 50-57

7 saf S, Yuval F, Uri K, et al. Google android: a comprehensive security assessment [J]. IEEE Security and Privacy, 2010, 8(2): 35-44

8 雷 蕾, 蔡权伟, 荆继武, 等. 支持策略隐藏的加密云存储访问控制机制 [J]. 软件学报, 2016, 27(6): 1432-1450
Lei Lei, Cai Quanwei, Jing Jiwu, et al. Encrypted cloud storage access control mechanism supporting policy hiding [J]. Journal of Software, 2016, 27(6): 1432-1450

9 刘 俊. 软件开发开放 API 接口的安全处理 [J]. 信息与电脑(理论版), 2017(8): 201-203
Liu Jun. Security Processing of open API interface for software development [J]. Information and Computer (Theoretical Edition), 2017(8): 201-203

10 刘树凯, 颜学雄, 王清贤, 等. 一种 Web 应用中轻量级的 JavaScript API 保护机制 [J]. 信息工程大学学报, 2018(2): 220-225
Liu Shukai, Yan Xuexiong, Wang Qingxian, et al. A lightweight JavaScript API protection mechanism in Web applications [J]. Journal of Information Engineering University, 2018(2): 220-225

11 王 玉, 宁可新, 朱蕾蕾. 基于 TLS 和 JWT 远程救助系统安全的 API [J]. 吉林大学学报(信息科学版), 2017, 35(6): 656-661
Wang Yu, Ning Kexin, Zhu Leilei. API based on TLS and JWT remote rescue system security [J]. Journal of Jilin University(Information Science Edition), 2017, 35(6): 656-661

12 郑志学. 基于防伪二维码技术的农产品溯源系统研究与开发 [D]. 株洲: 湖南工业大学, 2016
Zheng Zhixue. Research and development of agricultural product traceability system based on anti-counterfeiting two-dimensional code technology [D]. Zhuzhou: Hunan University of Technology, 2016

13 郑 雪. 基于 SOA 的面向商户第三方支付系统的设计与实现 [D]. 上海: 上海交通大学, 2014
Zheng Xue. Design and implementation of a third-party payment system for merchants based on SOA [D]. Shanghai: Shanghai Jiaotong University, 2014

14 周满元. 电子支付系统中双重签名的研究与应用 [J]. 四川大学学报(工程科学版), 2007(增刊 1): 196-199
Zhou Manyuan. Research and application of double signature in electronic payment system [J]. Journal of Sichuan University (Engineering Science Edition), 2007(S1): 196-199

15 姜建武, 李景文, 陆妍玲, 等. 基于 RESTful API 的智慧旅游系统设计与实现 [J]. 测绘与空间地理信息, 2017, 40(7): 57-61
Jiang Jianwu, Li Jingwen, Lu Yiling, et al. Design and implementation of smart tourism system based on RESTful API [J]. Surveying and Spatial Geography Information, 2017, 40(7): 57-61

Open API Security Authentication Mechanism Based on Request Body

JIANG Jian-wu, HU Yao, LI Jing-wen*
(College of Geomatics and Geoinformation, Guilin University of Technology, Guilin 541004, China)

[Abstract] Aiming at the problems of identity forgery phishing attacks, account and business information leakage and API platform malicious attacks, an API security authentication mechanism based on Request Body was proposed. The mechanism consists of five parts: double signature verification, request body encryption, URI verification, interface authority authentication and exception detection. It is used to prevent phishing scams, strengthen user data transmission security and improve the ability of the API platform to resist attacks. Through online testing and actual project verification, it shows that the mechanism can ensure the security of users and interfaces while ensuring the speed of API authentication.

[Key words] open API security certification restful API inforrmation encryption

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net