第43卷 第6A期
2016年6月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

云计算中身份认证技术研究

周长春[1] 田晓丽[2] 张 宁[2] 杨宇君[3] 李 铎[3]

(北京电子科技学院研科处 北京 100070)[1] (北京电子科技学院研究生院 北京 100070)[2]

(西安电子科技大学通信与工程学院 西安 710071)[3]

摘 要 对于云平台中用户之间的安全性认证问题,在分析 openstack 云平台的平台架构、安全认证组件 keystone、云计算中身份认证的主要安全性问题及当前云环境中主流的身份认证技术的基础之上,针对云平台下的统一身份认证机制及统一身份认证技术的漏洞,着重分析了 OpenID 身份认证的工作原理,提出了 OpenID 当前存在的安全性问题,并得出了一些改进方案。最后以 OpenID 改进技术为基础,在 openstack 平台上实现了身份认证技术。

关键词 openstack,身份认证,OpenID

中图法分类号 TP399 文献标识码 A

Research on Identity Authentication Technology in Cloud Computing

ZHOU Chang-chun[1] TIAN Xiao-li[2] ZHANG Ning[2] YANG Yun-jun[3] LI Duo[3]

(Research Division, Beijing Electronic Science and Technology Institute, Beijing 100070, China)[1]

(College of Postgraduate, Beijing Electronic Science and Technology Institute, Beijing 100070, China)[2]

(School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)[3]

Abstract For the security authentication between the user in question of cloud platform, based on the openstack cloud platform architecture, security authentication keystone components, the identity authentication of the main security issues in cloud computing and the current cloud environments mainstream identity authentication technology, aiming at the mechanism for unified authentication and unified identity authentication technology platform under the cloud of vulnerability, this article analyzed the work principle of OpenID authentication, presented currently existing OpenID security problems, and obtained some improvements. Finally on the basis of the OpenID improvement techniques, the identity authentication technology was realized on the openstack platform.

Keywords Openstack, Identity authentication, OpenID

云计算的主要任务是将大规模的计算资源或者存储资源通过用户终端的方式将服务提供给用户,使用户根据自己的需求直接获取计算机资源,而不用处理那些不必要的资源及存储管理,从而大大地降低了信息化的复杂程度。其中云计算的类型主要有私有云、公有云和混合型[1]。在实际的应用过程中,大多数采用的是以开源项目为主要技术核心的公有云或私有云建设。Openstack 是一个旨在为公共及私有云的建设与管理提供软件的开源项目。随着云计算的发展,越来越多的人开始关注对 openstack 的应用,目前 openstack 的大多数研究者和使用者都将它作为基础设施即服务(IaaS)资源的通用前端来进行研究与应用。但是现阶段云计算中所涉及的安全问题也多种多样,包括身份认证、访问控制、底层虚拟机之间的通信协议等,尤其是云计算的不断壮大所带来的海量访问及身份认证等问题,基于单一凭证的身份认证技术手段早已经不能满足用户对云安全的需要。本文基于这种现象,重点介绍了云计算中当前身份认证的主要技术,并进行了分析和比较;对 OpenID 身份认证技术提出了改进方案,并将其应用在 openstack 平台上。

1 云计算中身份认证技术的介绍

云计算是将计算作为基础设施而产生的一种基于交付和使用的模式,它既是一种计算模式,也是一种商业模式。它可以将大量的资源和服务部署在由计算机所构成的共享资源池上,并通过网络使用户能够以按需收费来从中获取计算、存储、网络等服务。云计算是基于并行计算、分布式计算、网络存储、虚拟化技术等网络技术和计算机技术发展延伸融合的一种产物。随着云计算的日益盛行,用户利用客户端(如浏览器)接入网络来便捷地使用其服务,因此作为云服务的第一道关卡,客户端接入网络时所要的身份认证技术成为了云计算安全领域研究中的关键问题。以下本文详细介绍了现阶段云计算中典型的身份认证技术。

1.1 基于 SAML 的身份认证

安全断言标记语言(Security Assertion Markup Language, SAML)是一种基于 XML 的面向 Web 服务的架构,可以用在不同的安全域来交换认证和授权数据[2]。在 SAML 标准中定义了两种角色:身份提供者(Identity Provider, IDP)

周长春(1963—),男,博士,研究员,主要研究方向为计算机网络、信息安全;田晓丽(1991—),女,硕士,主要研究方向为云计算安全性分析与信息安全,E-mail:1483646150@qq.com;张 宁(1993—),女,硕士,主要研究方向为大数据的安全性分析;杨宇君(1989—),男,硕士,主要研究方向为混淆密码的安全性分析;李 铎(1991—),男,硕士,主要研究方向为密码攻击。

和服务提供者(Service Provider,SP),两者构成了不同的安全域。它的主要功能就是实现在逻辑安全域中身份提供者和服务提供者之间的认证、授权信息的传输及断言形式的表达。SAML 规范体系主要由安全断言(Assertion)、请求/响应协议(Protocol)和绑定(Binding)[3]组成,其结构如图 1 所示。

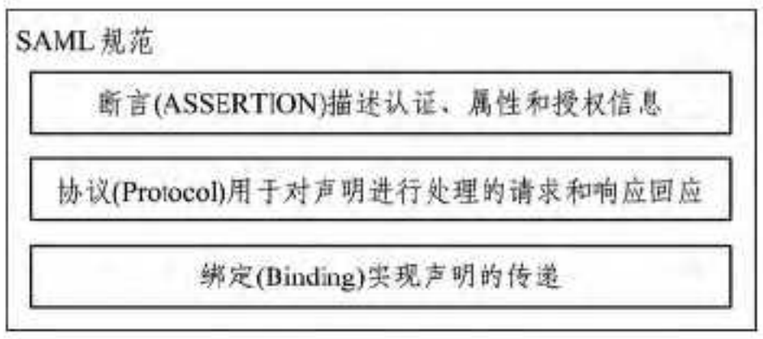

图 1 SAML 规范体系

断言的主要功能是判断所执行的操作是否允许主体对特定资源的授权等信息的访问。SAML 断言有 3 种类型:认证断言、属性断言以及授权断言。认证断言主要是确认用户在特定时间、特定机制下身份的验证;属性断言是对用户的特定属性信息的判断;授权断言确认用户访问资源的权利。SAML 最常用的协议是请求/响应协议,它的主要内容是规定 SAML 数据通信时的信息类别以及数据格式等问题,并且描述了断言是怎样在 SAML 请求(Request)和响应(Response)消息格式内打包的。SAML 中信息的通信主要依据具体协议的绑定来实现。对主体的查询有 3 种类型:认证查询、属性查询以及授权决议查询。绑定是将 SAML 请求和响应信息以一种标准消息的格式映射到 SAML 协议上。SAML 规范中给出的是与超文本传输协议(HTTP)和简单对象访问协议(SOAP)的绑定机制。图 2 所示为 SAML 断言在 SAML 请求和响应消息格式内的打包方式。SAML 断言直接嵌入 SOAP 信息头部,SOAP 再嵌入到标准的 HTTP 报文后进行传输。

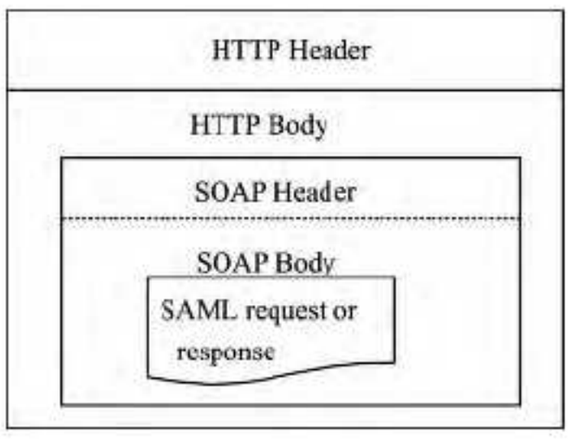

图 2 SAML 与 HTTP 和 SOAP 的嵌入式绑定

SAML 规范中定义了 3 种不同的角色:用户代理(通常为 Web 浏览器)、身份提供者即断言方(IdP)和服务提供者即信任方(SP)。在基于 SAML 的 SSO 方案中,用户通过用户代理向 SP 请求 Web 资源,用户以提交类似用户名、密码等能够证明其身份的信息来向 IDP 请求凭证。若用户的验证信息正确,便将其获取的 IDP 凭证及服务请求交给 SP 并产生一个 SAML 请求,请求的内容主要是要求 IDP 对获取的 IDP 凭证进行断言,IDP 自身进行判断并产生一个 SAML 断言给 SP。SP 从 IDP 处获取该身份断言后,会根据断言结果来决定是否为该用户提供服务[4]。

1.2 基于 OAuth 的认证授权管理

开放授权(Open Authorization,OAuth)最主要的任务是允许授权第三方网站访问它们存储在云平台上的信息,并且保护用户敏感信息不被外界披露的开放式身份认证方式。它是一个为用户资源的授权提供一个开放标准的联合协议,且支持细粒度的权限控制[5]。OAuth 的认证和授权的过程主要包括 3 个角色:服务提供者、用户和第三方,这里的第三方通常是网站。在 OAuth 授权过程中,OAuth 不是将用户认证凭证如用户名和密码直接提供给第三方,而是通过用户提供访问令牌给第三方网站的方式来提高协议的安全性。云平台中 OAuth 的认证授权过程如图 3 所示,用户使用第三方网站对云平台上的资源进行操作时,会通过第三方发起一个签名请求来获取一个临时令牌。云平台对第三方的身份进行验证后授予临时令牌,用户利用这个临时令牌和用户名密码等验证凭证授权给第三方网站,准许它可以访问所请求的资源。授权成功之后,第三方根据已经授权的临时令牌向云平台申请访问令牌,云平台确认第三方网站身份正确后给第三方发放访问令牌。第三方网站使用获取的访问令牌访问存放在云平台上的用户资源。

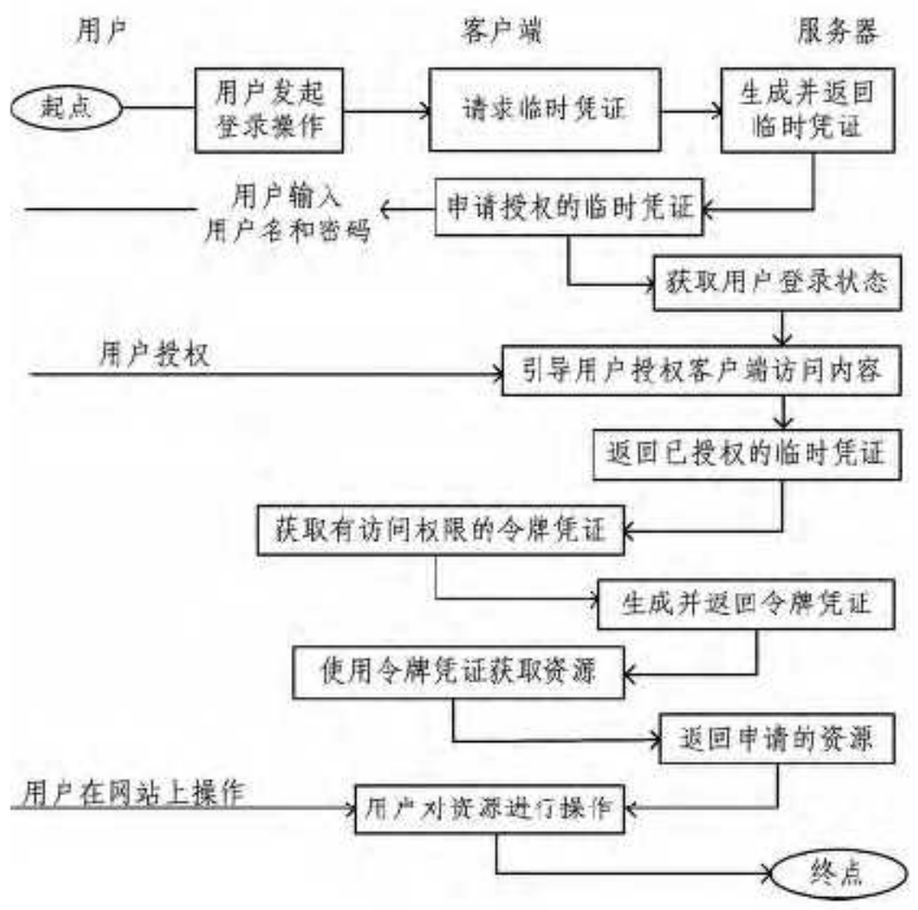

图 3 OAuth 认证授权过程

1.3 基于 OpenID 的身份认证技术

OpenID 是一个以用户为中心的身份识别框架,以 URL 来唯一标记用户身份,具有开放性、自由及分散等特征[6]。OpenID 身份认证技术是云计算中身份认证的主要应用技术。OpenID 的原理主要是用户通过拥有的 URL 在登录网站时作为自己的身份认证,而不是以用户名和密码的验证方式来进行用户身份认证。OpenID 系统主要有用户、OpenID 支持方(OpenID Relying Part,RP)(主要是支持用户用 OpenID 账号登录网站)、OpenID 提供方(OpenID Provider,OP)(主要是提供 OpenID 账号注册)、认证存储等服务[7]。其协议流程如图 4 所示。

图 4 OpenID 认证原理

在 OpenID 的认证过程中,用户首先要去 OpenID 提供方

网站注册一个包含用户名和密码的用户账号，以获得一个身份标识 URL。用户将代表自己身份的 URL 发送给 RP，RP 从接收到的 URL 中分析出 OP 的名称，并与之关联；并向 OP 发出认证用户身份的请求，RP 将用户重定向到 OP 服务器，并带上认证参数。OP 服务器直接从用户浏览器中读取 cookie，对用户进行认证。认证结束后，OP 会把认证响应返回给 RP，RP 收到响应后重新校验参数，检查认证结果。判断认证是否通过，以此确定是否允许该终端用户登录。

尽管 OpenID 和 OAuth 以及 SAML 认证技术已经被广泛地采用，但是这 3 种认证技术仍各自存在着一些问题还尚未解决。本文提出的认证方案主要对 OpenID 技术进行改进，旨在提供一种基于用户的安全性统一身份认证方案。

2 基于 OpenID 身份认证技术的改进方案

OpenID 作为一种高效的以用户为中心、支持跨域的身份认证架构，具有很多优点，对于用户来说，它有效地避免了重复的注册、填写用户信息等繁琐过程，使用户登录更加简单便捷，并且在 OpenID 身份认证的过程中，用户只需要注册一个 OpenID 账号就可以在任何支持 OpenID 的网站中自由登录，免去了记忆大量账号的麻烦，在一定程度上降低了用户密码泄露的风险[8]。然而，从安全性方面分析，OpenID 身份认证技术很容易受到网络钓鱼攻击。网络钓鱼很容易通过伪造成 Web 站点来欺骗用户，引诱用户在钓鱼网站上进行个人信息的登录，从而通过截取输入信息来获取用户的密码等一些个人信息。在身份认证和数据访问授权的过程中，用户每当被重新定向到身份提供商处进行身份验证时，就会面临着钓鱼攻击的威胁。在 OpenID 认证的过程中，RP 和 OP 在建立关联时最容易遭受到网络钓鱼攻击，因为它们在建立关联时，使用的是匿名 Diffie-Hellman 密钥协商，参与协商的一方都不知道另一方的具体身份。因此在遭受到攻击时，OP 或 RP 都很难发现遭受了网络钓鱼。

防范钓鱼网站的关键在于让 OpenID 服务器和用户之间可以双向认证识别，使用户能够识别出目标网站并能够区分出哪些是通过精心设计，与目标网站非常相似的钓鱼网站，从而抵御网络钓鱼。其实现流程如图 5 所示。在认证之前，用户在 OpenID 服务器上注册账户信息时，除平常需要填写的用户名、密码等基本的个人信息外，认证码和初始个人标识符也作为用户身份的凭证。之后 OpenID 服务器会利用 MD5 密码算法或其他算法对用户在登录时填写的认证码进行加密或提取摘要等处理，并将其处理结果存入 OpenID 服务器数据库中。当 RP 将用户重定向到 OP 登录页面时，用户首先需要输入注册时所用的验证码，然后 OP 对所提取的消息摘要与数据库中所存储的消息摘要进行对比。若核对通过，OP 就会显示用户上次登录时输入的个人标识，并且判断与上次登录时输入的个人标识是否相同，若相同则继续操作，否则认定其为钓鱼网站。认证之后，用户输入登录密码和标识符，OP 核对之后，若无误就会向 RP 返回响应，经过认证之后用户的标识符也会被存储在数据库中，用来提供下一次用户对 OpenID 服务器的认证。这就是 OP 与用户完整的双向认证过程[9]。

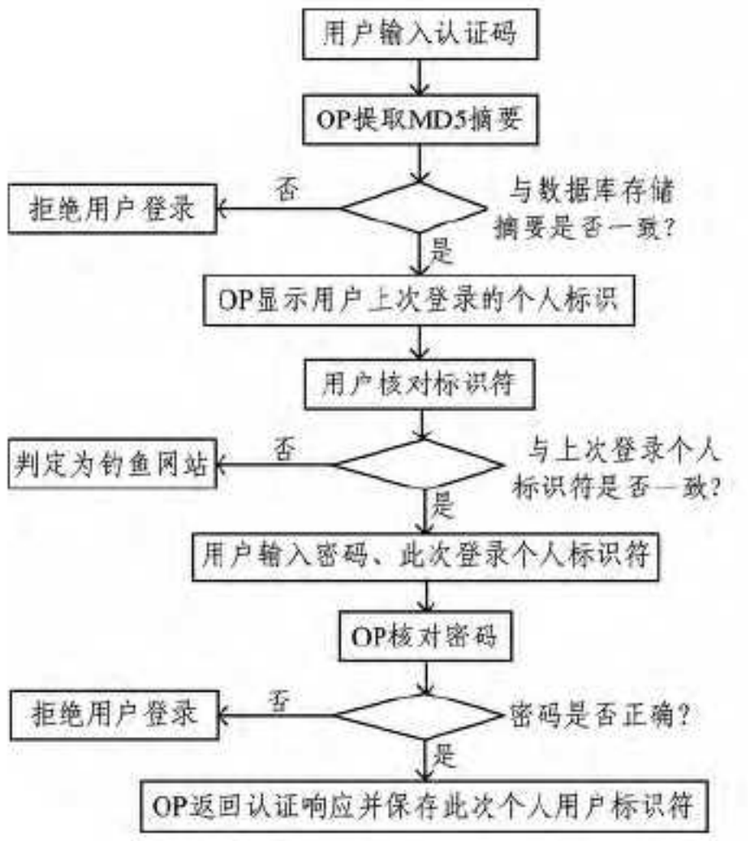

图 5 用户与 OpenID 服务器的双向认证流程

3 OpenID 身份认证技术在 Openstack 平台上的实现

Openstack 的组件之间的调用都需要经过 Keystone[10]，它集成了用于身份验证、策略管理和目录服务的功能[11]。当用户拿着有效的用户名和密码去 Keystone 认证后，Keystone 会使用签名密钥和数字证书来签名用户的 Token，根据服务目录、用户角色以及元数据应用 CMS 来生成 CMS Token 并返回给用户。用户发起一个 API 请求时，会将 CMS Token 一起发送过来。而每个 API 端点都会持有一份 Keystone 的拷贝，包括签名证书、撤销列表、CA 数字签名。API 端点使用这些字节直接验证用户的请求，从发起的请求中抽取 CMS Token，通过验证 Token 签名、过期日期以及撤销列表来判断 Token 是否有效，有效则进行请求处理，无效则驳回请求[12]。其认证过程如图 6 所示。

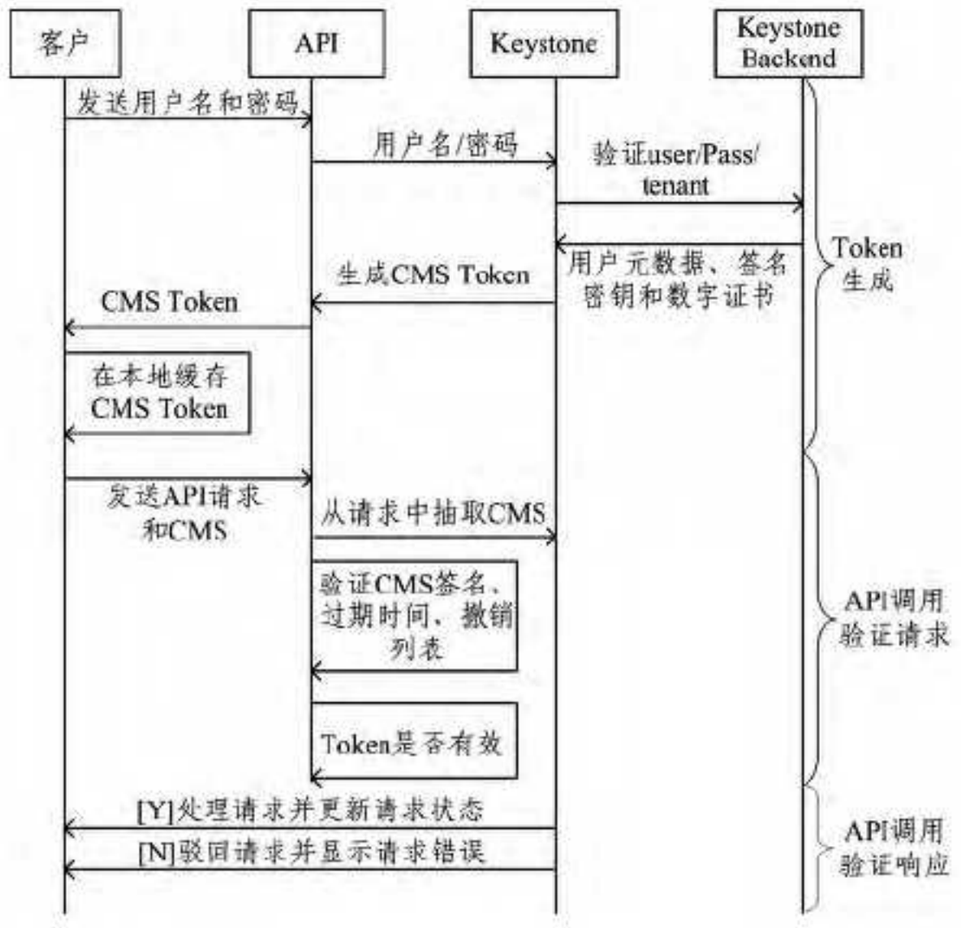

图 6 PKI Token 过程

OpenID 认证服务在 openstack 中的应用更能增加其认证的安全性，OpenID 认证在 RP 中主要涉及两个阶段[13]：OP 对端口 URL 的发现及原始信息的检索，验证从 OP 中所得的 URL。因此可以把 OpenID 认证技术分为两个阶段[14]，每一个阶段都会调用不用的 API（认证请求 API 和认证验证

（下转第 369 页）

mation Science,2011,192(3):111-121

[42] Khan S U, Lavagno L, Pastrone C. Online Authentication and Key Establishment Scheme for Heterogeneous Sensor Networks [J]. International Journal of Distributed Sensor Networks, 2014,2014:1-11

[43] Lee J H,Kwon T,Ehlers F. Location-Aware Key Management for General Deployment of Wireless Sensor Networks [J]. International Journal of Distributed Sensor Networks,2014,2014:1-17

[44] Dai H,Xu H. Key predistribution approach in wireless sensor networks using LU matrix [J]. IEEE Sensors Journal,2010,10(8):1399-1409

[45] 马春光,张秉政,孙原. 基于按对平衡设计的异构无线传感器网络密钥预分配方案 [J]. 通信学报,2010,31(1):37-43

[46] Zhang C X,Cheng L L,Wang X D. Efficient key pre-distribution protocol for Heterogeneous wireless sensor networks [J]. Journal of Computational Information Systems, 2013, 9(11): 4583-4592

[47] 覃荣华,解永生,袁晓兵. 异构分组无线传感器网络密钥管理机制 [J]. 华中科技大学学报(自然科学版),2012,40(4):19-42

[48] 掌明,王锁萍,徐鹤. 基于分簇的无线传感器网络动态密钥管理方案 [J]. 南京邮电大学学报(自然科学版),2012,32(1):98-103

[49] 钟晓容,马春光. 一种抗 LU 攻击的传感器网络密钥预分配方案 [J]. 计算机学报,2013,36(6):1155-1167

[50] 胡小春,陈燕,梁俊斌,等. Sink 移动的无线传感网中高连通性密钥预分配方案研究[J]. 数学的实践与认识,2014,44(6):128-134

[51] 黄廷辉,杨旻,崔更申,等. 基于 LEACH 协议的无线传感器网络密钥管理路由方案[J]. 传感技术学报,2014(8):1143-1146

[52] 李兰英,易春焕,孙建达,等. 基于单位元的无线传感器网络密钥管理方案[J]. 计算机工程与应用,2015(2):94-98

(上接第 341 页)

API)。其认证过程如图 7 所示。用户会对 GUI 服务器提供一个认证码的 OpenID 认证请求。OP 服务器通过调用 API 服务器来提取消息摘要,并与数据库存储的消息和上次个人登录标识进行对比,若一致则不是钓鱼网站,否则可能是钓鱼网站。若确定安全,前端会调用认证请求 API,并返回需要重定向的所有 OpenID 参数。GUI 服务器会分析信息并向 UA 发送重定向命令。UA 会重新定向 OP,并进行验证,一旦验证成功,OP 会把 UA 重定向到 GUI 服务器,并调用认证验证 API 对所有的过程进行验证,成功就会允许用户登录界面,否则就失败

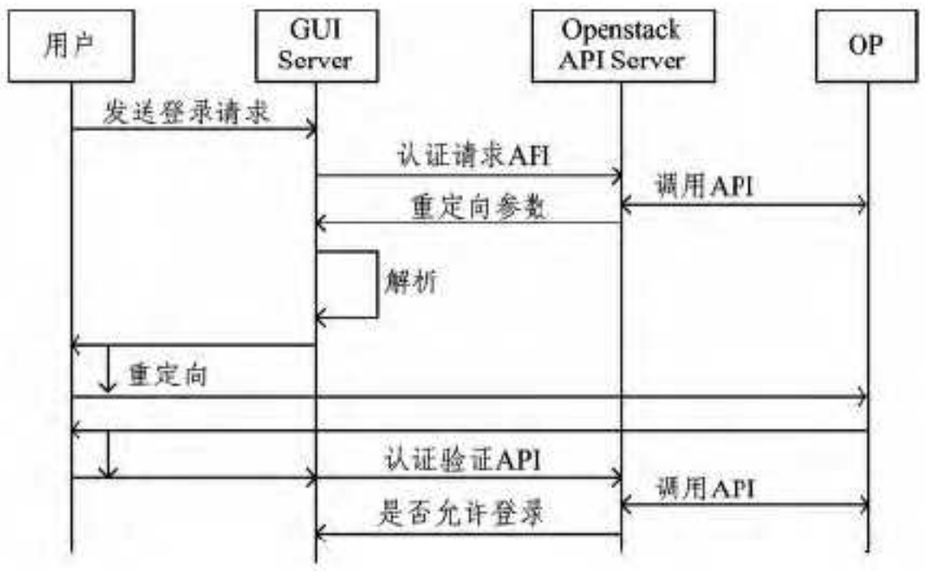

图 7 openstack 中 OpenID 的认证流程

结束语 安全是信息化时代最重要的问题,随着云计算的发展,如何保障信息安全不被泄露,也是云计算进一步发展所必须要考虑的问题。身份认证作为访问云计算资源的第一步,身份认证的安全性问题是当前云计算安全领域最重要的问题,其研究也受到了各界的广泛关注,近年来也取得了一定的进展。本文主要对当前云计算中主流的身份认证技术 SAML、OAuth 和 OpenID 做了大致的介绍,重点讲解了身份认证机制原理并在此基础上提出了 OpenID 身份认证当前的一些缺陷,同时做了一些改进。对 openstack 中的认证组件 Keystone 框架进行了深层解析,通过对其对象模型和认证原理的研究,分析出了当前 Keystone 组件的安全性问题,并将 OpenID 身份认证改进方案应用到 openstack 中,更进一步增加了其安全性。

参 考 文 献

[1] Hu Luo-kai, Ying Shi, Jia Xiang-yang, et al. Towards an Approach of Semantic Access Control for Cloud Computing[C]// Cloud Computing, 2009. Beijing, China: Springer Berlin Heidelberg,2009:145－156

[2] OASIS Standard. SAMLV2.0[EB/OL]. (2005). http://docs.oasis-open.org/security/saml/v2.0

[3] 江浩浩,徐东升. SAML 在集成身份认证中的应用[J]. 电信网技术,2012(7):17-21

[4] 王群,李馥娟,钱焕延. 云计算身份认证模型研究[J]. 电子技术应用,2015,41(2):135-138

[5] 江伟玉,高能,刘泽义,等. 一种云计算中的多重身份认证与授权方案[J]. 信息网络安全,2012(8):7-10

[6] 秦晓娜,郝平,何恩. 基于 OpenID 安全认证的 Web 实时通信系统[J]. 信息安全与通信保密,2013(4):70-72

[7] 夏晔,钱松荣. OpenID 身份认证系统的等级模型研究[J]. 微型电脑应用,2011,27(4):20-23

[8] Wei J,Zhang M,Ding X,et al. Research on Multi-Level Security Framework for OpenID[C]// International Symposium on Electronic Commerce & Security,2010. 2010:393-397

[9] 吴志勇,孙乐昌. 针对钓鱼攻击的防范技术研究[J]. 信息安全与通信保密,2006(11):126-128

[10] 张进铎,毛承国,李硕,等. Openstack 开源云平台主模块的架构分析[J]. 信息化技术与信息化,2014(4):244-247

[11] Sitaram D, Phalachandra H L, Vishwanath A, et al. Keystone Federated Security[C]// ICITST. 2013:659-664

[12] 熊微,房秉毅,张云勇,等. OpenStack 认证安全问题研究[J]. 邮电设计技术,2014(7):21-25

[13] Khan R H,Ylitalo J,Ahmed A S. OpenID Authentication As A Service in OpenStack[C]// Internation Conference on Information Assurance & Security,2011. 2011:372-377

[14] Chadwick, David W, Matteo C. Security APIs for My private cloud-granting access to anyone, from anywhere at anytime[C]// Third IEEE International Conference on Cloud Computing Technology and Science,2011. 2011:792-798