基于Yale-CAS的单点登录的设计与实现

沈 杰,朱程荣
(同济大学 计算机科学与技术系,上海 200331)

摘 要:随着网络信息化建设的加快,各种企事业单位依托网络进行数字化办公,用户经常遇到不同的系统,以至于必须来回切换登录。单点登录(Single Sign-On)就是为解决传统认证机制中存在的问题而提出的一种技术。在分析比较常见的单点登录技术的技术上,实现了基于Yale-CAS的单点登录系统,并针对原有系统缺点进行了改进,使得认证和授权分离,并且减轻了认证服务器的负担,减少了网络传输,方便用户快速访问资源。

关键词:SSO;单点登录;CAS;认证;授权
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2007)12-0144-03

Design and Implementation of Single Sign-on Using Yale-CAS

SHEN Jie, ZHU Cheng-rong
(Dept. of Computer Science and Technology, Tongji University, Shanghai 200331, China)

Abstract: With the population of network and the more quick information construction of enterprise, many workers in company and government use network to wok, user often want to access different application server, this lead up to logging in server many times. SSO (single sign-on) is put forward to resolve the difficult situation in the tradition authentication mechanism. Analyses and compares some SSO techniques, and realizes an SSO system using CAS, and also improves the old system, separates the authentication and authorization. It makes the authentication server's burden less, and decreases the transmission in the network, let people access resources more quickly.

Key words: SSO; single sign-on; CAS; authentication; authorization

0 引 言

随着信息技术和网络技术的不断发展,企业内部的Web应用系统不断增多。例如某企业拥有财务系统、人事系统、顾客关系系统、采购系统等等,这些系统对于提高企业运行效率起了不可忽视的作用,但是也暴露了不少问题,传统的架构方法在每个应用中采用了以各自服务为中心的认证系统,即每个登录系统只负责一个应用。为此用户必须记住每一个系统的用户名和密码,这给用户带来了不少麻烦,特别是随着系统的增多,出错的可能性就会增加,受到非法截获和破坏的可能性也会增大。软件设计人员也不得不在每创建一个新的应用时也必须为其创建新的登录系统,造成了资源浪费。基于安全和效率的双重考虑,用户迫切需要一种更新的认证机制。单点登录(Single Sign-on,简记SSO)[1,2]的概念随之产生出来。

SSO是一种方便用户访问多个系统的技术,用户只需在登录时进行一次注册,就可以在多个系统间自由切换,不必重复输入用户名和密码来确定身份。单点登录将原有分散的用户管理集中起来,各个系统依靠相互信任的关系来进行用户的身份认证。

SSO具有以下优点:

(1)更优的管理控制:对应每个用户的权限与特权,仅有一个授权列表。

(2)更高的用户工作效率:用户不至于再陷入多次登录的麻烦,也不用再为访问网络资源要记住多个密码。

(3)更高的网络安全性:所有可用的SSO方法均提供了安全身份验证,并提供了对用户与网络资源的会话进行加密的基础。取消多个密码,还减少了安全漏洞的普遍来源——用户总喜欢写下的密码。

1 常用SSO技术的分析

1.1 基于凭证的认证技术

这种技术的思想是在认证服务器与系统之间通过凭证建立起某种信任关系。用户在访问应用之前,首先需要登录认证服务器,通过验证取得凭证,通过凭证

收稿日期:2007-02-07
作者简介:沈 杰(1982-),男,浙江嘉兴人,硕士研究生,研究方向为网络安全及电子政务;朱程荣,副教授,研究方向为容错计算与信息安全。

用户就可以访问认证服务器所能够管理的各个应用系统,用户不需要其他额外的认证就可以访问。Kerberos 就是基于凭证的典型应用:Kerberos 作为一种可信任的第三方认证服务,是通过传统的密码技术(如:共享密钥)执行认证服务的。

这种技术实现应用广泛,但不足之处在于系统实现比较复杂,往往需要附加额外的软件安装配置,这对于多应用系统的整合比较困难。

1.2　基于 Web 请求代理的认证技术

这种技术仅限于 Web 应用。其思想主要是建立认证服务器代理对于 Web 应用的所有认证请求,对于已经认证的用户认证服务器只是转发 URL 到应用系统。

基于这种技术的单点登录系统,用户首次访问应用系统的 Web 请求将被转移到认证服务器,认证成功,则将用户的合法信息包含在原有请求中,一并转发给应用系统。

基于 Web 请求的代理也需要在认证服务器和应用之间建立信任关系,但这种技术的实现比较简单,对于原有系统的改动比较少。大多数单点登录系统采用这种技术。

1.3　基于二次登录的认证技术

这种技术的思想是,也要建立一个认证服务器,这个服务器中包含原有各个应用系统的认证系统副本,用户登录认证服务器后,对于应用的访问首先会从认证服务器中取得对应应用系统中该用户的密码等认证信息,然后自动进行二次登录。

这种技术较前两种灵活,对原有系统的改动比较少,但是加重了认证服务器的负担,所有的认证信息都在服务器上保留,增加了安全隐患,而且因为各自系统保留了自己的认证系统,使得认证信息在应用系统和认证服务器之间的同步成了一个问题。

2　Yale－CAS 协议基本原理

CAS(Central Authentication Service)是由耶鲁(Yale)大学开发的认证系统,提供可信任的服务来使应用系统认证用户。目前该系统仍处于进一步的开发中。

2.1　Yale－CAS 协议认证过程

CAS 中心认证服务被设计成一个独立的 Web 应用。它通常可以在几个流行的 Web 服务器中运行,例如 Tomcat,Web logic 等等。

用户可以通过以下三个 URL 访问:

login URL, validation URL 和可选择的 logout URL[3]。

CAS 的认证过程见图 1。

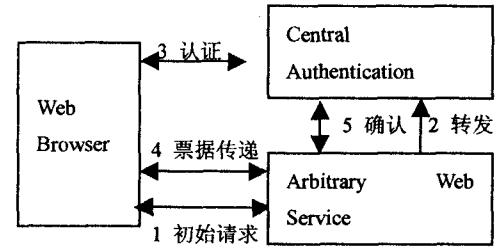

图 1　CAS 认证过程

2.2　若干概念

* TGC: Ticket Granting Cookie,用户访问 CAS 服务器的凭证;私有的保护的 cookie;对用户透明可重复的 ticket。

* ST: Service Ticket,浏览器访问应用的凭证;对用户透明不可重复的 ticket;短期有效性。

2.3　认证细节

1) 取得 TGC 的过程。

用户想访问应用(图 2(下同)过程 0),但因为没有 ST,更没有 TGC,所以用户被转到 CAS 认证页面(过程 1),提供用户名(ID)和密码之后,用户得到 TGC(过程 2)。

2) 取得 ST 的过程。

在取得 TGC 之后用户访问应用(图 3(下同)过程 3),用户通过 TGC 拿到访问应用的 ST(过程 4,5)。应用收到用户的 ST,发到 CAS 验证,并返回结果(过程 6,7)。应用返回用户的访问资源(过程 8)。

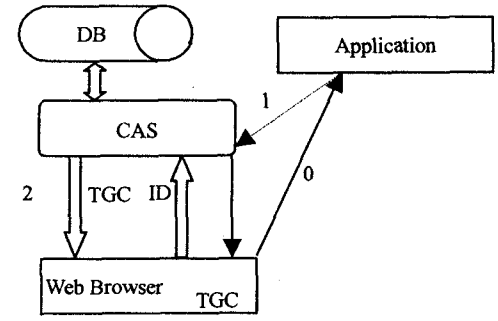

图 2　取得 TGC 过程

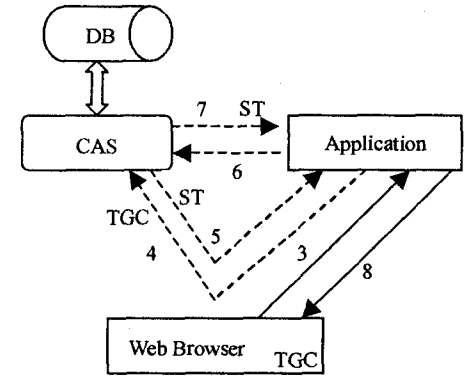

图 3　取得 ST 的过程

3 系统设计

3.1 原有CAS系统设计存在的问题

通过分析Yale-CAS协议工作原理,发现CAS设计存在以下若干问题:

(1)认证服务器负担过重。因为原有的系统认证内容不统一,在认证过程中需要将各个系统的认证信息全部写入session,信息量太庞大,加重网络负担。

(2)系统不灵活。若要增加新的应用,则需要重新改写认证模块,加入新应用的认证信息。

(3)授权和认证结合,系统层次不清,容易产生误操作,不方便今后系统添加新应用。

3.2 基于CAS的设计改进方案

设计一个基于Yale-CAS的单点登录系统[1,4~6],由三部分完成单点登录的实现:CAS中心认证服务器(CAS Server),提供用户认证和应用授权管理;CAS应用端代理(CAS Client),截取用户对应用系统的访问请求,与CAS Server通信来验证用户,与应用系统通信传递用户登录票证;用户信息库,认证服务器端后台数据库,保存用户身份信息及权限信息。新系统保持原有CAS系统中取得TGC的过程(见图3),但对取得ST和授权访问应用部分做了修改(见图4)。图4中左数据库中只存用户认证信息,不包含授权信息。而右数据库只是保留原有应用系统中的授权部分而将认证部分转移到了左数据库。

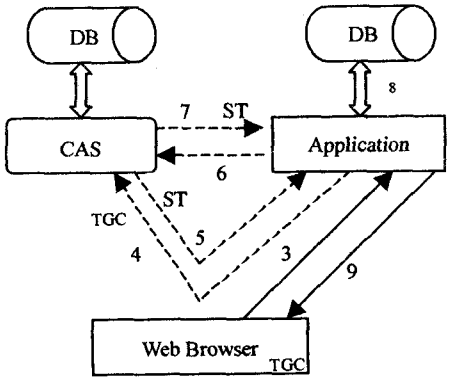

图4 改进系统流程

针对原有CAS系统缺点,做如下改进:

1)改进认证服务器中的认证模块。

原有认证模块不仅写入公有的认证信息,而且会将各个应用的特定信息(包括授权信息)全部写入session,所以去掉原有的多余信息,只在session中写入公有的认证信息。

2)改进认证客户端。

在每个客户端中增加过滤,使得在认证服务器送回ticket、客户端接收到之后,紧接着根据所访问的应用确定要写入哪个应用的session数据(见图4过程8),并将授权功能分布于各个应用。

用户使用系统时,首先发出访问应用请求,应用代理截取用户请求,检查用户有无登录票证;如果没有,将认证转向认证服务器CAS Server,提示用户登录。CAS Server接受用户登录信息,从后台数据库中读取用户信息,与用户登录信息进行比较,验证通过则为用户生成一个全局会话(session),并向CAS Client返回登录票证。CAS Client将登录票证交给用户请求的第三方应用系统;应用系统与CAS Server通信确认票证有效性,验证后获得CAS Server返回的用户标识信息;应用系统根据此信息实现对用户的授权。用户在登录票证有效期内可持续访问第三方应用;也可以通过注销等操作,放弃票证,结束访问。

经过以上登录过程后,用户访问参与单点登录的其它应用时,由中心认证服务器上的用户会话信息生成新的应用票证实现用户与应用的认证,用户无需再次参与认证过程,实现"一次登录,多方认证"的单点登录过程。

新系统最大限度地减少对了原有应用的改动,保持了应用系统中的授权功能,将授权功能从认证服务器中分离,也符合软件工程中松耦合的思想。

新系统保持原有CAS的以下认证特点:

(1)CAS的双向认证过程结束时,认证中心返回的票证TGC中包含用户身份信息,应用程序可以在不用访问用户的密码的情况下验证用户身份。

(2)在用户浏览器接受Cookie的情况下,认证过程中生成的Cookie可以重新对CAS确认用户。用户不用再次输入身份标识和密码。

(3)利用CAS可实现用户应用的单点登录。用户在向CAS认证后,CAS将用户重定向到所请求的应用服务。

(4)CAS提供了一个用户访问退出的URL,用户可以利用它来退出CAS,取消票证TGC。

4 基于CAS的SSO系统的实现

在实际运行过程中,实现了三个应用之间的单点登录。

4.1 前期准备

将原有三个应用系统中的用户进行集中、统一管理。在系统数据库中存储用户信息,包括身份认证信息和用户个人基本信息。用户可以使用SSO系统的统一身份访问多个系统;现有应用系统的账户整合。

4.2 认证客户端模块的实现

将对应用的访问加入到CAS系统的统一认证范

(下转第150页)

ples of distributed computing. New York, NY, USA: ACM Press, 2001: 274 - 283.

[6] Craver S. Zero - knowledge Watermark Detection[C]// Proceedings of the Third International Workshop on Information Hiding, Lecture Notes in Computer Science 1768. Berlin: Springer - Verlag, 2000: 101 - 116.

[7] Zou X X, Dai Q, Huang C, et al. Zero - Knowledge watermark verification protocols [J/OL]. Journal of Software, 2003, 14(9): 1645 - 1651. http://www.jos.org.cn/1000 - 9825/14/1645.pdf.

[8] HE Yong - Zhong, WU Chuan - Kun, FENG Deng - Guo. Publicly Verifiable Zero - Knowledge Watermark Detection[J/OL]. Journal of Software, 2005, 16(9): 1607 - 1616. http://www.jos.org.cn/1000 - 9825/14/1606.pdf.

[9] Beth T. Efficient zero - knowledge identification scheme for smart cards[C]// Advances in Cryptology: Proceedings of Euro - crypt'88. New York, USA: Springer - Verlag, 1988: 77 - 84.

[10] Blum M, Feldman P, Micali S. Non - interactive zero - knowledge and its applications (extended abstract)[C]// In Proceedings of the 20th Annual ACM Symposium on Theory of Computing. [s.l.]: ACM, 1988: 103 - 112.

[11] De Santis A, Persiano G. Zero - knowledge proofs of knowledge without interaction[C]// In 33rd Annual Symposium on Foundations of Computer Science. Pittsburgh, Pennsylvania: IEEE, 1992: 427 - 436.

[12] Bellare M, Micali S, Ostrovsky R. Perfect zero - knowledge in constant rounds[C]// In Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing. Baltimore, Maryland: [s.n.], 1990: 482 - 493.

[13] Brassard C, Crepeau C, Yung M. Constant - Round Perfect Zero - Knowledge Computationally Convincing Protocols[J]. Theoretical Computer Science, 1991, 84: 23 - 52.

[14] Goldreich O, Micali S, Wigderson A. Proofs that yield nothing but their validity or all languages in NP have zero - knowledge proof systems[J]. J. ACM, 1991, 38(3): 690 - 728.

[15] Koblitz N. Elliptic curve implementation of zero - knowledge blobs[J]. Journal of Cryptology, 1991, 4: 207 - 213.

[16] Almuhammadi S, Sui N T, McLeod D. Better Privacy and Security in E - Commerce: Using Elliptic Curve - Based Zero - Knowledge Proofs[C]// 2004 IEEE International Conference on E - Commerce Technology (CEC'04). Washington, DC, USA: IEEE Computer Society, 2004: 299 - 302.

[17] Mao Wenbo. Modern Cryptography: Theory and Practice[M]. Beijing: Publishing House of Electronics Industry, 2004.

[18] Balasubramanian R, Koblitz N. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes - Okamoto - Vanstone Algorithm[J]. Journal of Cryptology, 1998, 11(2): 141 - 145.

(上接第146页)

围,并代理与认证服务器的通信。并且在各个应用中加入filter,使得当用户在认证服务器验证,被转移到应用之后,应用系统能够根据自身原有的授权机制,授予用户特定的权限。这样就将授权的功能分布于各个应用之中。

4.3 认证服务端模块的实现

认证服务端主要处理统一的用户认证、用户代理访问、代理访问验证以及用户请求服务的验证;含两个认证子模块,分别用于处理基于密码的身份验证和基于服务的身份认证。原有系统中只需要改进子模块基于密码的身份验证。将身份的验证通过与数据库中的用户数据做比较,判断是否合法。

5 结束语

文中提出的基于CAS的SSO系统的设计与实现,完成了统一的身份认证,实现企业门户系统的单点登录,提高了用户使用系统的效率,减轻了系统管理员的工作负担,并且使得认证和授权分离,克服了原有系统的缺点,系统的分工以及层次清晰,在实际运行过程中,实现了三个应用之间的单点登录,稳定性好,能够快速反应,对于今后企业门户系统实现单点登录有较强的参考价值。

参考文献:

[1] 谭立球,费耀平,李建华.企业信息门户单点登录系统的实现[J].计算机工程,2005,31(17):102 - 104.

[2] The Open Group. Single Sign - On[EB/OL]. 2005. http://www.opengroup.org/security/sso, 1995 - 2005.

[3] JA - SIG Central Authentication Service[EB/OL]. 2006. http://www.ja - sig.org/products/cas/, 2005 - 2006.

[4] 邱 航,杜向辉.单点登录原型系统KSSO的设计与实现[J].计算机工程与设计,2006,27(9):1645 - 1648.

[5] Zhao Gang, Zheng Dong, Chen Kefei. Design of single sign - on[C]// E - Commerce Technology for Dynamic E - Business, 2004. IEEE International Conference. Beijing, China: [s.n.], 2004: 253 - 256.

[6] 金 辉. Single sign - on[EB/OL]. 2002 - 10. http://www - 900. ibm. com/developerWorks/cn/security/se - sso/index.shtml.