基于 Web 的 RBAC 权限机制在电子病历系统的研究

董 斌[1],陈进哲[1],刘秀玲[2],张 瑜[3]
DONG Bin[1],CHEN Jin-zhe[1],LIU Xiu-ling[2],ZHANG Yu[3]

1.河北大学附属医院 计算机中心,河北 保定 071000
2.河北大学 电子信息工程学院,河北 保定 071000
3.保定市卫生局 医政处,河北 保定 071000
1.Computer Center,Hebei University Affiliated Hosptial,Baoding,Hebei 071000,China
2.Department of Electronic and Informational Engineering,Hebei University,Baoding,Hebei 071000,China
3.Department of Medical Management,Baoding Health Bureau,Baoding,Hebei 071000,China
E-mail:dbin2000@163.com

DONG Bin,CHEN Jin-zhe,LIU Xiu-ling,et al.Research on privilege system for role-based access control based on Web in electric medical record system.Computer Engineering and Applications,2008,44(26):233-235.

Abstract: This paper firstly analyzes the necessity of secure authentication for the electric medical record system,and then researches the Cookie technology emphatically,discusses the application system of secure Cookie in detail,and analyzes the secure character of secure Cookie with the technology of PKI digital authentication.Further,with the research of role-based access control,this thesis puts forward an effective secure system based on Web.This system utilizes the secure Cookie to guarantee the security,in the mean time,introduces the cache system to improve the efficiency of achieving privilege.Through the syntheses experiment with actual data,it is proved that this system has enhanced the efficiency in some degree.Finally,the present thesis analyzes the application of this system in the electric medical record system with its characters.

Key words: role-based access control;secure Cookie;cache system;Public Key Infrastructure(PKI);electric medical record system

摘 要: 首先分析了电子病历系统安全验证的必要性,着重研究了 Cookie 技术,探讨了安全 Cookie 的应用机制,并且通过使用 PKI 数字认证技术,分析了安全 Cookie 的安全特性。进一步通过对基于角色的访问控制的深入研究,提出了一种基于 Web 高效安全机制。该机制在利用安全 Cookie 保证安全性的前提下,引入缓存机制来提高权限获取效率。通过实际数据模拟实验,结果表明该机制在效率上有一定程度的提高。最后,针对电子病历系统的特点分析了该机制在其中的应用。

关键词: 基于角色的访问控制;安全 Cookie;缓存机制;公钥基础设施(PKI);电子病历系统
DOI: 10.3778/j.issn.1002-8331.2008.26.071 **文章编号:** 1002-8331(2008)26-0233-03 **文献标识码:** A **中图分类号:** TP393

随着卫生信息化的不断发展,电子病历成为了数字化医院的核心。围绕电子病历发生的争论也越来越激烈[1]。同时电子病历的法律效力和地位也成为社会关注的热点。针对电子病历系统的现状,探讨了基于 Web 的高效安全机制在电子病历系统中的有效性和应用前景。

1 基于角色的访问控制

Web 环境下,应用基于角色的访问控制 RBAC(Role Based Access Control)模型[2-3]主要有两种结构[2]:一种是 User-pull 结构,一种是 Server-pull 结构。同时,实现这两种模型来获得用户属性时,还可以使用基于用户的或是基于主机的认证信息模型。下面重点介绍 User-pull 结构。

在 User-pull 结构中,用户从角色服务器那里得到自己的角色,然后在把这些角色提交给 Web 服务器,其中的认证信息既可以是基于用户的,也可以是基于主机的,它之所以被称为 User-pull 结构,是因为用户从角色服务器得到自己的角色,同时这些角色也正是在角色服务器被指派给域中的用户的。用户与服务器之间的通信是在使用 HTTP 协议部署的 Web 浏览器和 Web 服务器之间进行的。

2 安全 Cookie

2.1 Cookie

由于 HTTP/1.1 协议本身是无状态的协议,为了方便用户的访问,维持服务器和客户端的连接状态,人们引入了 Cookies 机

基金项目: 河北省教育厅科研计划项目(the Science Project of the Hebei Education Department under Grant No.2005349)。
作者简介: 董斌(1980-),男,助理工程师,主要研究领域为医疗信息系统、信息安全;陈进哲(1965-),男,工程师,主要研究领域为医疗信息系统的开发与设计;刘秀玲(1977-),女,讲师,主要研究领域为信息系统安全与设计;张瑜(1980-),女,政工师,主要研究领域为医疗信息系统的开发与设计。
收稿日期: 2007-11-06 **修回日期:** 2008-01-28

制[4]。Cookies 是包含用户相关信息编码的文本串。通常在用户访问使用 Cookies 的 Web 站点时,通过浏览器发送到用户的内存并且在浏览器关闭后保存到硬盘中。下次用户再次访问时,Web 服务器可以获取这些 Cookies 并从中得到用户的信息。

所有的 Cookies 具有相同的基本结构,一个典型的 Cookies 结构具有以下几个字段:Cookie_Name 和 Cookie_Value 包含了一个 Web 站点想要保留的信息,Name_Cookie 相应的值为“Daemon”;Date 是 Cookie 的有效生存期;Domain 则是该 Cookie 在哪个主机或域名内有效;Flag 则指明是否所有在指定 Domain 内的主机都可以访问该 Cookie 的信息;Path 限定该站点内只能在指定路径下才能使用此 Cookie; 如果 Secure 标志是 True,那么该 Cookie 将只能在安全信道(如 SSL)内进行传输。

通过对一般 Cookie 的研究[5],发现它存在的安全性威胁主要有以下 3 点:

(1)网络安全威胁(Network security threats):由于 Cookies 在网络中是以明文方式传输,网络路由中的任何主机都可以截获并篡改 Cookies 的内容。

(2)终端系统的威胁(end-system threats):Cookies 在客户端也是以明文保存的,这就存在用户自由修改 Cookies 内容的可能。

(3)Cookies 收获攻击威胁(cookie-harvesting threats):由于有以上两个威胁的存在,网络中的用户就可以收集网络中的 Cookies 并把它们用于假冒他人身份攻击。

2.2 安全 Cookie

安全 Cookie 针对上面提到的 3 种潜在的攻击,提供了 3 种安全服务:认证、完整性和机密性。认证服务实现对 Cookie 所有者的验证;完整性服务保护 Cookie 免受非授权的修改;机密性服务保护 Cookie 的值对非授权用户的不可见性。

2.2.1 基于数字签名的认证

如果 Web 服务器知道用户的公钥,那么如 DSA 或 RSA 这些数字签名技术可以使用 Cookie 验证用户。要是用这种方法,用户需要额外的浏览器软件来声称一个包含签名时间戳的 Cookie。例如,当 *A* 要访问一个知道它公钥的 Web 服务器时,*A* 的计算机产生一个时间戳并生成一个如图 1 的 Sign_Cookie,在这个 Sign_Cookie 中包含有用 *A* 的私钥对时间戳的数字签名。当 *A* 连接到该 Web 服务器时,它接受到 *A* 的 Sign_Cookie 并用 *A* 的公钥来验证该签名。

Domain	Flag	Path	Cookie_Name	Cookie_Value	Secure	Date
IS.hbu.edu.cn	True	/	Sign_Cookie	Signature	False	1/8/2005

图 1 认证 Cookie(Cookie 的数字签名)

2.2.2 完整性

在如图 2 所示的这组安全 Cookie 中,Life_Cookie 的“Cookie_Value”字段存储了安全 Cookie 的生命期(截止日期),使得 Web 服务器可以在安全 Cookie 有效时才检查这组安全 Cookie 的完整性。即使浏览器只向服务器发送了相关的“Cookie_Name”字段值和“Cookie_Value”字段值(通过域字段和标志字段的筛选),Cookie 的声明服务器在域中仍可以和 Web 服务器生成一个策略来检查其他字段的完整性。例如:如果一个策略规定“Domain”、“Flag”、“Path”和“Secure”字段的值分别为“Is.nbu.edu.cn”、“True”、“/”和“False”,那么 Web 服务器就可以利用这些值来检验 Cookie 的完整性。

	Domain	Flag	Path	Cookie_Name	Cookie_Value	Secure	Date
Name_Cookie	Is.hbu.edu.cn	True	/	Name_Cookie	Bob	False	12/10/2005
Role_Cookie	Is.hbu.edu.cn	True	/	Role_Cookie	Student	False	12/10/2005
Life_Cookie	Is.hbu.edu.cn	True	/	Life_Cookie	1/30/99	False	12/10/2005
Pswd_Cookie	Is.hbu.edu.cn	True	/	Pswd_Cookie	hashed_pswd	False	12/10/2005
Key_Cookie	Is.hbu.edu.cn	True	/	Key_Cookie	encrypted_key	False	12/10/2005
Seal_Cookie	Is.hbu.edu.cn	True	/	Seal_Cookie	Seal_of_Cookies	False	12/10/2005

图 2 安全 Cookie

3 基于 Web 的 RBAC 高效安全机制的实现[6]

3.1 安全性

在一个应用 RBAC 域中,当用户想要在 Web 服务器上执行操作时,首先用户要连接到角色服务器,并提交认证信息。当用户通过了身份认证之后,角色服务器将会在“用户-角色”指派数据库中查询并将结果指派给用户, 同时生成一组安全 Cookie。随即这组安全 Cookie 将安全地发送给用户且保存在用户的计算机中。在这组安全 Cookie 到达终止期之前,用户无需再次访问角色服务器得到角色的指派,也可以在该域中安全的使用其 Role_Cookie。然后当用户浏览器中输入 Web 服务器(其中存有“权限-角色”指派信息)的地址提出访问要求时,浏览器把相应的一组安全 Cookie 发送到 Web 服务器。Web 服务器利用认证 Cookie(如 IP_Cookie 或 Pswd_Cookie)对用户身份认证,将 Cookie 中的值和用户提交值进行比对。最后,Web 服务器通过角色服务器的公钥对 Seal_Cookie 中角色服务器的数字签名进行验证来判断这组 Cookie 的完整性。如果所有的 Cookie 都是有效的且顺利通过验证, 那么 Web 服务器将信任 Role_Cookie 中的角色信息并通过“权限-角色”指派约束用于 RBAC 模型。这样就有效地防止了对角色信息的攻击行为(如假冒、篡改等),提高了角色信息的安全性,进而保证了基于 Web 的 RBAC 模型的安全性。

3.2 高效性

在 RBAC 模块中,假定角色、权限的基本信息都存储在相关介质中。权限角色指派、角色层次关系以及限制等其他关系都以某种形式被存储起来。由于角色层次关系的存在,权限可以通过角色继承间接对应某些角色,角色还可以通过角色继承间接继承其它角色。

在一个实际的访问控制系统中,一般来说权限的数量比较大,角色的数量不大,但是却有着十分复杂的继承关系。一个中等规模的访问控制系统中的角色模型可以有深达 8 层的继承关系。在这种情况下,给定了一个角色和一种权限,判断某角色是否具有该权限可能是一个很深的递归查找过程,这是整个权限管理中最消耗资源的部分。在这种需要反复查询的情况下,引入缓存机制可以极大地提高后续查找的效率,从而提升系统的整体性能。

3.3 模型实现

3.3.1 创建安全 Cookie

为某一具体应用创建安全 Cookie 的流程如图 3 所示:当用户通过 Web 浏览器连接到该域的角色服务器(支持 HTTP 协议)时,首先出现域登陆界面(由 HTML 语言编写的表单)要求输入用户的 ID 和密码。Set_Cookie 模块先找到用户 ID、密码

和用户机 IP,然后同认证数据库中记录的用户信息进行比对,当通过认证后就可以将“用户-角色”指派数据库中对应与用户 ID 的角色指派给该用户。随后 Encryption 模块将用验证 Cookie 的 Web 服务器的公钥对用户输入的密码进行加密,并由 Set_Cookie 模块将密文保存在 Pswd_Cookie 中。接着,Set_Cookie 模块将根据得到的数据生成 IP_Cookie、Pswd_Cookie、Name_Cookie、Life_Cookie 和 Role_Cookie,并将对应的值赋予相应的字段:IP_Cookie 存储用户机 IP;Pswd_Cookie 存储由 Encryption 生成的密码密文;Name_Cookie 存储用户名;Life_Cookie 存储 Cookie 的生命期;Role_Cookie 存储由 Set_Cookie 模块指派的角色。

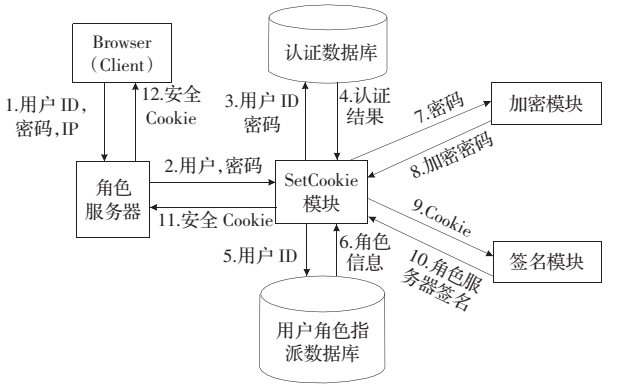

图 3　创建安全 Cookie 流程图

生成了上述 Cookie 之后,Set_Cookie 模块将调用 Signature 模块来保证 Cookie 的完整性。Signature 模块利用角色服务器的私钥来对 IP_Cookie、Pswd_Cookie、Name_Cookie、Life_Cookie 和 Role_Cookie 一起加密,将密文返回给 Set_Cookie 模块。Set_Cookie 模块得到密文后,随即生成 Seal_Cookie,并将得到的密文保存在 Seal_Cookie 中。

最后,角色服务器将生成的安全 Cookie 和 HTTP 响应头一起发给用户浏览器,并把 Cookie 存储在用户机里。其中 IP_Cookie、Role_Cookie、Name_Cookie 以明文形式出现,Pswd_Cookie 中保存的是用 DES 算法加密的密码的密文,而 Seal_Cookie 中保存的是对上面的 Cookie 利用角色服务器的私钥使用 RSA 算法得到的数字签名(由于结果在中文操作系统下显示因此为不可理解的乱码;如在英文操作系统下,该字段结果为无规律的字符串)。

3.3.2　验证安全 Cookie

如图 4 所示,在应用 RBAC 的域中,当用户连接到 Web 服务器(该服务器可以接收安全 Cookie)时,这个链接被重定向到 Login 模块。相应的安全 Cookie 被发送到 Web 服务器,而用户浏览器则弹出登陆表单要求用户输入用户 ID 和密码。Login 模块验证所有 Cookie 的有效性,比对两个 IP 地址:一个来自于 IP_Cookie,另一个来自于环境变量 REMOTE_ADDR。如果两个 IP 相一致,那么用户就通过了基于主机的认证,并且将隐藏变量 STATE 的值置为“ip_true”以表明用户通过认证;如果两个 IP 不一致,那么用户的请求将被服务器拒绝。

当用户通过了 Login 模块的认证后,将转到 Verify_Pswd 模块。Verify_Pswd 模块首先检查隐藏变量 STATE 以确定用户是否成功通过了上一模块的认证。如果隐藏变量 STATE 的值为“ip_true”,那么 Verify_Pswd 模块将用 Web 服务器的私钥来解密 Pswd_Cookie(其中保存的是用 Web 服务器公钥加密的用户密码)。然后 Verify_Pswd 模块将解密的明文和用户的密码进

图 4　验证安全 Cookie 流程图

行比对,如果二者一致,那么基于用户的认证也就通过了,同时将隐藏变量 STATE 的值置为“Pswd_true”以表示该阶段已经完成;如果二者不一致,那么用户或者重新输入密码,或者到角色服务器去获取新的 Cookie。

在通过了 Verify_Pswd 模块的验证后,将调用 Verify_Signature 模块来检验 Cookie 的完整性。与 Verify_Pswd 类似,Verify_Signature 模块首先检查隐藏变量 STATE 以确定用户时候成功通过了密码认证阶段。当隐藏变量 STATE 的值为“Pswd_true”时,该模块将用角色服务器的私钥来验证 Seal_Cookie 中的签名的完整性。如果通过了完整性验证,就说明这些 Cookie 不曾被修改过,而且隐藏变量 STATE 的值被置为“All_true”以表示通过了这个模块的验证;如果完整性验证失败,那么用户必须重新开始,而且需要重新向角色服务器请求新的 Cookie。最后,系统从 Role_Cookie 中提取出角色用于下面的 RBAC 模块以获得用户的权限。

3.3.3　缓存机制

在本文所描述的机制中,在前一部分已经实现了安全的获取用户角色的模块,因此在设计缓存机制时,仅考虑对“角色-权限”指派数据进行缓存。这种缓存是一种细化的缓存策略,添加缓存的时机是在查询用户权限对应关系的时候。算法流程为:首先查找缓存中是否存在该“角色-权限”指派,如果存在则算法返回 True;否则从该角色开始向下查找其子角色是否存在该“角色-权限”指派,如果存在且已经保存入缓存中则返回算法 True;如果存在而未保存入缓存,则将所有可以继承该角色权限的“角色-权限”指派添加入缓存,并算法返回 True;如果该角色不具有这个权限同时它可以继承的子角色也不具有该权限,则算法返回 False。

4　实验和结果分析

对于缓存机制的效果,本文设计了一个实验模型来进行对比实验。

硬件环境:主机配置 CPU P4 2.6 GHz,内存 256 M,硬盘 80 G 5 400 r/min;软件环境:Delphi7.0+SQL Server2000;实验数据:本医院电子病历系统日志,时间 2005 年 10 月 19 日 16:00 至 2005 年 10 月 20 日 16:00(实验数据保存在.log 文件中,文件大小 772 Mb);应用实验模型,使用实验数据中 17:00 至 18:00 的数据进行了几组实验,具体实验结果如表 1 所示,其中实验用时表示将所有实验数据都运算完成的时间,性能提高量为缓存容量改变后实验用时减少率(与缓存容量为 0 相比而言)。

通过实验模型的运行结果,可以得出如下结论:

(1)引入缓存机制可以有效的提高 RBAC 模型中获取权限的效率,其效率的提升可以达到 30%左右,因此具有一定的应用价值;

(下转 238 页)