Web 信息系统 资源访问控制 事务处理 (6)

计算机科学 1999 Vol. 26№. 8

2-29

Web信息系统中的资源访问控制*)

The Access Control Of Resources in a Web-based Information System

TP399 TP393

刘亚霄 刘卫东 徐 恪
(清华大学计算机科学与技术系 北京 100084)

Abstract Information accesses control in a web-based information system is one of the security bases of the system. In this paper, we introduce an algorithm describing a new method of organizing information resources and a way for browsing and publishing them. These methods can be applied for a web-based, multi-user system. Some means of web authentication are also given in this paper.

Keywords Access control, Web based information system, Role-oriented, Information browsing, Authentication

0 引 言

基于 Web 的事务处理系统中,用户所获取的信息应该是可控的,他们只能获取自己权限范围内所能获取的资源。在一个系统中,如何使用户的视野受到严格的控制是权限管理的关键问题。

在普通的 Client/Server 结构的 MIS 系统中,用户的权限控制是通过定义用户可以获取和修改的资源来加以控制的[4],在基于 Web 的事务处理系统中,也可以采用同样的思路。

在 Web 上,所有的信息都是通过 HTML 主页表达在用户的浏览器上的。根据 HTML 的特征,我们把用户资源分为三类:

1. 表示资源。这是构成 HTML 主页的主体资源。包括文字、图像、声音、动画以及表示信息的 Java Applet 等等;

2. 处理资源。在HTML 中的 Form 是这类资源的主体。通过 Form,用户可以输入自己的请求,经过 CGI 的处理以后,就可以根据用户的意愿获取他们所有能够获取的信息。在近期的 Web 应用中,Java 等客户端处理语言的出现,进一步增强了最终用户的 Web 信息处理能力;

3. 连接资源。Web 上的主页都是具有其链接的,URL 是它们的表达形式。上述两种资源可以通过主页上的 Anchor(锚)表示和其它的主页链接在一起,用户使用浏览器就可以方便地从一个信息库访问到另外的一个信息库(在 Form 中的提交按钮等也是一种隐性的连接资源)。

前 1,2 两种资源,我们称之为基本资源。对于最终用户所能看到的一个完整的在浏览器中表达的主页,我们称之为一个节点(Node)。在系统管理人员看来,每一个节点都是上述三种资源的组合体。为了限制用户的访问范围,就要考虑下述的几个问题:

1. 在系统中,同一个节点上的所有的资源的权限域都有可能不同。例如,有些用户访问图书销售主页时,因为他们不是销售协会会员,他们就不能够看到提供会员优惠的 Form。相反地,他们应该看到对会员有优惠的广告。会员则正好相反。而对于两者来说,他们都应该能够看到书籍简介。

2. 在主页上的连接是有可能泄漏信息的。例如,如果在主页上有一个会员才能访问的联谊会的主页的连接的话,非会员点击这个连接通常会得到他没有这个权限的警告,他就会了解系统中有这个部分,站点信息就有泄漏的可能。

3. 在浏览器中,用户通过身份认证后,可能会记录下这个主页的 URL,也就是利用浏览器中的书签(Bookmark)方式。以后,当他失去访问这个主页的权限时,他有可能通过这个书签重新访问这个主页,造成非法信息泄漏。如果仅仅利用 Web Server 的密码保护特性保护身份又会造成每页都要求用户输入密码,使系统实用程度下降。

下面我们将就 Web 事务处理系统的权限管理提出一种组织方法以及该方法的具体实现。

1 Web 信息权限管理

1.1 信息的表示

首先定义一下信息表示的基本方法。对于上述的表示资源(也是主要的信息模块)我们称之为"项

*)本课题是 CERNET"九五"重点项目—"教育部全国普通高校招生信息系统"的子课题,受 CERNET 资金资助。

目"(Item);处理资源我们称之为"模块"(Module);连接资源我们称之为"链"(Link);Module 和 Link 的连接目标我们称之为"目标"(Target)。这样每个有意义的节点就可以表示成为由至少一个 Module 或 Item 与可能有的其它资源组成的集合:

E={Module};F={Item};L={Link};

Node={((e)|(f))|((e)*|(f)*|(l)*)*,e∈E,f∈F,l∈L};

一个表示示例如下:

```
Node A
{
    Item 1
    Link 1
        Target Node B# Item X
    Item 2
    Module 2
        Target Node C
}
```

作为一个 Link 的 Target,可以是一个 Node,也可以是一个 Item(表示为 Node N# Item I)。一个 Module 的 Target 只能够是一个 Node。每一个 Item 都是一些最基本的 HTML 表示方法,例如,Item 1 可以表示如下:

```
Item 1
{
    〈H1〉欢迎访问本站点〈/H1〉
    〈BR〉
}
```

1.2 主题

主题(Subject)是资源的组织形式,每一个主题由子主题构成,子主题既可以是主题,也可以是基本资源(不包括连接资源):

Subject=((e)|(f)+Subject)|((e)|(f));

主题之间访问路径按照继承关系进行组织:如果一个用户拥有对于某个子主题的访问权限,那么他就拥有对这个子主题的所有子主题的访问权限,主题之间一般的访问途径是树的结构,即不允许有主题访问的回路出现。

1.3 角色

角色是访问权限的定义域和用户管理的操作域,每个主题的访问权限是面向角色分配的,也就是说,角色是分配权限的基本单位[5],用户是最基本的登录单位,所有的用户都至少属于一个角色。同一个用户可以属于多个角色,从而可以拥有不同的访问权限。角色是管理用户权限之间从属关系的基础。角色之间对主题的访问权限可以交叉,也就是说不同的角色都可能拥有对于一个主题的访问权限,角色之间没有明确的依赖关系。角色由数据管理人员根据需求建立。

1.4 基于角色的主题信息浏览

在面向角色的组织中,用户所能够得到的最终结果是一个节点中的资源子集,当用户访问到某一个节点时,系统首先查找如前面所描述过的节点的资源信息,然后根据每一个信息的类型进行逐条处理,确定访问权限。给定基本算法如下:

```
Procedure Information_Access(Current_Node, Current_User)
Begin
    While 还有资源描述没有处理完 Do Loop
        Case 当前资源描述属性 Do
            Item:Begin
                判断用户的访问权限 Author(Item,Current_User)
                If 用户拥有访问权限 Then 获取此 Item 的 HTML 描述和相应的 Web 资源,加入最终表示界面中
                End if
                End
            Module:Begin
                判断用户的访问权限 Author(Module, Current_user)
                If 用户拥有访问权限 Then
                    获取连接资源的节点信息
                    进行目标节点权限子查找:
                        Target_Auth(Target, null,Current_User)
                    If 子查找成功 then
                        获取此 Module 的 HTML 描述和相应的 Web 资源,加入最终表示界面中
                    End if
                End if
                End
            Link:Begin
                获取连接资源的节点信息
                如果形式是 TargetNode 进行目标节点权限子查找:Target_auth(Target,null,Current_User)
                如果形式是 TargetNode# TargetItem 进行目标节点权限子查找:Target_auth(Target,TargetItem,Current_User)
                If 子查找成功 then
                    获取此 Link 的 HTML 描述和相应的 Web 资源,加入最终表示界面中
                End if
            End
    获取下面一条描述信息
    End Loop
End
```

对于当前资源权限判断算法如下:

```
Function Author(Resource, Current_user) Returns Boolean
Begin
    R:=Current_user 所属的角色集合
    I:=能够访问 Resource 的角色集合
    If RII=∅ then//用户没有访问权限
        Return False
    Else
        Return True
    End if
End
```

对于目标节点权限的子查找算法如下：

```
Function Target_auth(Target_Node, Target_Item, Current_User)Returns Boolean
Begin
    If Target_Node 被标记为已查找 Then
        Return False
    End if
    If Target_Item is not null then
      //目标节点表示形式是 TargetNode#Item 形式
      判断用户对 TargetItem 的访问权限 Author(TargetItem,Current_user)
      If  用户没有访问权限 then
          Return False
      Else
          Return True
      End if
    Else
        //目标节点表示形式是 TargetNode 形式
        标记此 Node 为已查找
        Not Found=True
        While TargetNode 的描述信息未处理完成,且 Not Found Do Loop
            Case 处理资源 Do
            Item:判断用户访问权限 Author(Item,Current_user)
                 If  有访问权限 Then
                     Return True
                 End if
            Module:判断用户访问权限 Author(Module, Current_user)
                 If  有访问权限 Then
                     判断用户是否对连接资源有访问权限 Target_auth(TargetNode,null,Current_user)
                 If  子查找成功 then
                     Ruturn True
                 End if
                 End if
            Link:判断 Target 类型,判断用户是否对连接资源有访问权限,
                 如 TargetNode#TargetItem
                 Target_ auth (TargetNode, TargetItem,Current_user)
                 如 TargetNode Target_auth (TargetNode, null, Current_user)
                 If  子查找成功 Then
                     Return True
                 End if
        处理下面一条记录
        End Loop
        Return False
    End if
End
```

2 身份认证实现方法

在系统中，身份的确认是一个重要问题，获取正确的用户访问身份，是进行权限控制的基础，在我们所研究的全国高校远程录取系统中，我们采用了如下的身份认证和身份传递机制。

系统是基于 Oracle 数据库的网络应用，使用的是 Oracle Web Server 2.1作为 Web 服务器，所有的 Node 描述信息、Item 和 Module 的表示信息以及主题权限分配和用户角色分配信息都是基于数据库进行管理的。

系统中的身份验证是通过询问用户身份代号和用户口令实现的，这主要是考虑到在 Web 应用中，目前的情况下采用集中认证方式比较困难。用户的 ID 和口令经过核实后，和用户登录的时间一起组成一个字符序列，然后采用 MD5 算法[6]进行演算，得到用户一次登录的唯一标识 MLogID，对于用户的 ID，我们也采用 IDEA 算法[7]进行加密处理，并替换掉密文中的非法字符，以此作为用户的身份传递标识--MUserID。对于 MLogID，一方面我们将其存储在数据库中，另一方面以 Cookie 形式存储到用户的客户方，不设定失效时间，当用户浏览器退出时自动失效。用户身份传递标识 MUserID 是访问每个节点的必备信息。在访问每个页面前，都通过用户标识解析用户身份，根据登记日志获取 MLogID，然后根据用户是否具有合法的 Cookie 来判断其是否合法访问。通过 Bookmark 访问中间页的用户将因为没有 Cookie 而被拒绝访问。管理人员通过定时查看日志可以避免一些其它的身份认证问题。

小结 基于 Web 的事务系统的权限管理是保证这个系统正常运行的重要机制。使用上述的机制可以通过编制 Item 并归结到某个主题快速的发布消息，并通过其它控制手段获得较好的效果。

参 考 文 献

1 Samarati P, et al. An Authorization Model for a Distributed Hypertext System. IEEE Trans. on Knowledge and Data Engineering, 1996,8(4):555～562

2 沈卫忠. 基于 Intranet 的管理信息系统 [清华大学硕士学位论文]. 1997

3 徐恪. 基于对象技术的安全信息系统的研究与设计：[清华大学硕士学位论文]. 1998

4 欧阳明. 基于 Web 的分布式信息系统的研究与实现：[清华大学硕士学位论文]. 1998

5 Tari Z, et al. A Role-Based Access Control for Intranet Security. IEEE Internet Computing, 1997(1)

6 Rivest R L. The MD5 Message Digest Algorithm, RFC 1320,1992

7 Lai X, Massey J. A proposal for a New Block Encryption Standard. In: Advances in Cryptology-Eurocrypt'90 Proc., New York: Springer-Verlag, 1990