a Way to Control Access by ASP

用 ASP 实现的一种权限访问控制

程爵平　刘 谨　(上海大学　200072)

摘要：ASP 技术由于编程简单、功能强大，并且安全性较好，使它成为一种主流的网络开发语言，本文用一个实例论述了用 ASP 实现的权限访问控制方法。

关键词：ASP　权限

1 前言

当前随着 Internet 的发展和企业信息化的实施，越来越多的企业开始用网络来建立自己的电子商务或信息管理平台，通过在 Internet 上发布产品信息，或构建自己企业的 Intranet 网络，企业可以达到对外沟通用户，提供新型的网上服务，对内实现管理手段的信息化，优化企业组织结构，提高企业对市场的灵敏度和竞争力。

在众多的网络开发软件中，ASP（Active Server Pages）技术由于编程简单，并且可以通过 ADO 等多种方式方便地访问网络数据库，而必将取代复杂的 CGI（公共网关接口）和繁琐的 Perl 编程语言，成为 Web 开发的主流语言。ASP 是微软公司提供的一种开放式的动态 Web 服务器应用程序开发技术，代表了微软公司开发技术的一个新发展，它将脚本、超文本和强大的数据库访问功能结合在一起，并提供了众多的服务器内置组件用于创建分布式的和基于 Web 的商业化应用程序。在为企业开发内部使用的 Web 信息管理平台时，为防制不相关的人访问或修改企业的机密信息，维护企业信息的安全性，有必要采用某种访问控制策略。下面用一个开发实例说明利用 ASP 实现的一种简单权限访问控制。

2 实例剖析

2.1 系统的结构框图

该系统是用 ASP 为某企业开发的基于 B/S 结构的 MIS 系统，它包括备品备件仓库管理模块、五金仓库管理模块、固定资产管理模块、设备综合管理模块等几个部分，公司为每一个模块都配有一名专职管理人员负责该模块数据的输入、修改和管理，公司的高层管理人员、财务人员或采购人员根据工作需要具有对某一个或几个模块浏览的权限。如图 1 所示：

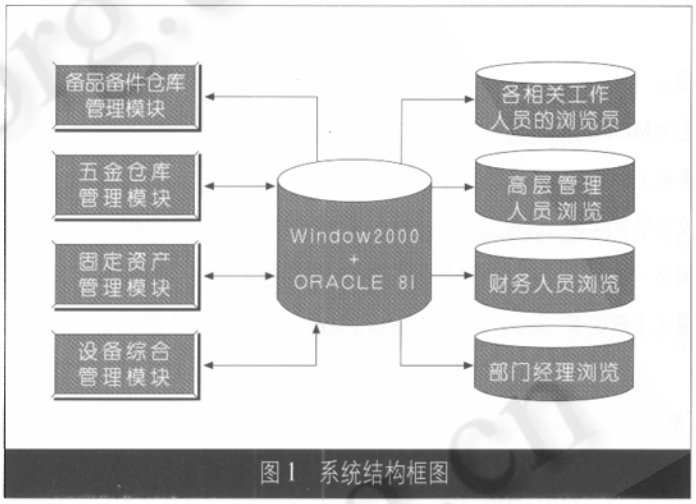

图 1　系统结构框图

2.2 基于角色的权限分配策略

对用户访问的控制主要是通过对用户权限进行控制和管理，权限是让用户对某一对象进行某一操作的权利，在本系统中这种权利是指用户查看页面或通过页面修改数据的权利。为了便于系统管理人员对用户权限进行管理，减少对用户授权的次数，我们采用了基于角色的权限分配策略，把一组系统级和对象级权限授给一个角色，一旦用户获得该角色，则自动拥有该角色所具有的所有系统级和对象级权限，同时根据实际需要系统管理员还可以将角色所不具有的对某一模块的权限赋给特定的人员。这种关系可以用图 2 来表示：

对图2所示的每个模块我们设定三种权限分别是无权限，即用户不能访问该模块（用 "0" 表示）、只读权限，用户只可以浏览模块中的数据但不能

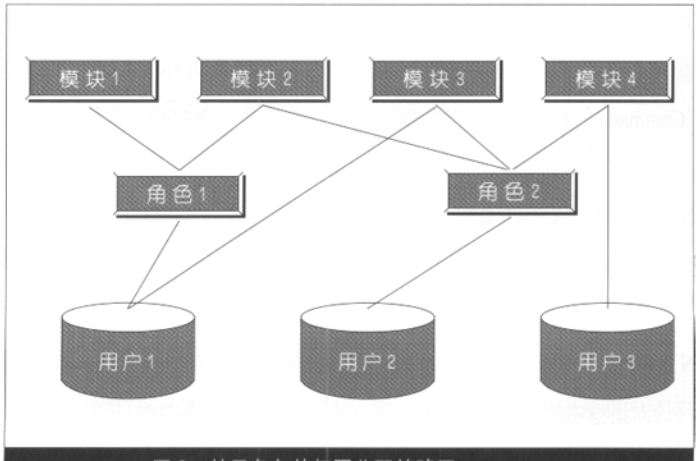

图 2　基于角色的权限分配策略图

表1

字段列名	数据类型	长度	描述
模块ID	NUMBER	12	PRIMARY KEY
模块名称	VARCHAR	40	NOT NULL

用户定义的角色表用来存放用户根据需要定义的角色，它包括角色ID，角色名称，字段结构和意义如表2。

表2

字段列名	数据类型	长度	描述
角色ID	NUMBER	12	PRIMARY KEY
角色名称	VARCHAR	40	NOT NULL

角色所具有的权限表存放分配给角色的权限，包括角色ID，模块ID及权限等字段，字段结构和意义如表3。

表3

字段列名	数据类型	长度	描述
索引ID	NUMBER	12	PRIMARY KEY
角色ID	NUMBER	12	NOT NULL
模块ID	NUMBER	12	NOT NULL
权限	VARCHAR	1	NOT NULL

用户表包括用户名、密码和状态字段，字段结构和意义如表4。

表4

字段列名	数据类型	长度	描述
用户名	VARRCHA2	15	PRIMARY KEY
登录密码	VARRCHA2	15	NOT NULL
用户状态	VARRCHA2	1	NOT NULL

用户所具有的权限表存放分配给用户的权限，它包括用户名字、模块名称、权限，字段结构和意义如表5。

表5

字段列名	数据类型	长度	描述
索引ID	NUMBER	12	PRIMARY KEY
用户名	VARRCHA2	15	NOT NULL
模块ID	NUMBER	12	NOT NULL
权限	VARRCHA2	1	NOT NULL

用户所具有的角色表存放分配给用户的角色，它包括用户名字和角色ID，字段结构和意义如表6。

表6

字段列名	数据类型	长度	描述
用户名	VARCHAR2	15	PRIMARY KEY
用户角色ID	NUMBER	12	NOT NULL

修改数据（用"1"表示）和读写权限，用户的最高权限允许用户修改数据（用"2"表示），管理员可以将这些模块的权限组合起来赋给所定义的角色，然后再将它授予某一类的用户，则该用户自动拥有该角色所具有的对模块的所有权限，也可直接将一个模块的权限授予某个用户。

为实现这种权限访问控制，我们首先需要在数据库中建立几个表，用以存放模块信息、用户定义的角色、授给角色的权限、用户信息、授给用户的权限和授给用户的角色等，具体表结构及各项意义如左所示。

模块信息表主要是为了系统能有更好的扩展性，它包括模块ID和模块名称二个字段，字段结构和意义如表1。

2.3 用ASP实现权限访问控制

ASP有两个重要的特性：其一是ASP是运行在服务器端的脚本语言，当客户端用户用浏览器向服务器申请访问基于ASP的脚本时，浏览器向Web服务器发出http请求，Web服务器通过ISAPI接口调用ASP脚本解释运行引擎（ASP.DLL），解释运行引擎将从文件系统或内部缓冲区获取指定的脚本并解释执行，最终处理结果形成HTML文件格式返回给浏览器，而不是将ASP的源文件直接下载给客户端，这样客户端用户不可能通过IE得到ASP的"源代码"，而只能看到ASP处理后返回的结果。其二，ASP提供了一个内置的存储在服务器上的Session对象，当客户端用户第一次访问ASP页面时，服务器便会为用户产生一个唯一不重复的Session误别码SessionID，在客户端，浏览器将这个SessionID存入到本地的Cookie中，并在下次访问服务器时提交给Web处理程序，Web处理程序根据这个SessionID找到服务器中以前存储的信息并使用它，该对象使得我们可以使用以前页面保存的数据，例如，可以保存登录页面用户输入的帐号和密码，在以后的每一页无需用户再输入就可以直接使用它。这两个重要特性是我们利用ASP实现权限访问控制的关键。

用户在登录页面中输入用户帐号和密码，提交后系统首先用一条语句

"select * from qx_userlog where username = '" & username & "' and password = '"&

password &"' and enabled = '1'"从用户表qx_userlog中查询用户是否存在，并且是否是有效的用户，如果存在则进行下一步操作，否则阻止用户登录系统。如果用户存在则要判断用户所具有的权限并将它赋给Session对象，以便系统在以后的页面中用以判断权限，赋值形式如下：

session("username") = username

session("WJ_PRIVIL") = name_mdlname2privil(session("username"), WJ_MDLNAME)

```
session("BB_PRIVIL") = name_mdlname2privil(session("username"),
BB_MDLNAME)
session("SZ_PRIVIL") = name_mdlname2privil(session("username"),
SZ_MDLNAME)
session("SY_PRIVIL") = name_mdlname2privil(session("username"),
SY_MDLNAME)
```

函数name_mdlname2privil () 的作用就是根据用户名和模块名称来返回用户对该模块相应的权限。源程序如下：

```
<%
function name_mdlname2privil(username,mdlname)
    if username = "" or mdlname2mdlid(mdlname) = "" then
        name_mdlname2privil = "0"
        exit function
    end if

    dim localsqlII,localrsII,tmp1,tmp2
    set localrsII = server.createobject("adodb.recordset")
        localsqlII = "select * from qx_userprivil where username = '" & username
& "'and mdlid = '"& mdlname2mdlid(mdlname)&"'"
        localrsII.cursorlocation = 2
        localrsII.open localsqlII,conn,1,3
    if(localrsII.eof) then
        tmp1 = "0"
        else
            tmp1 = localrsII("privil")
        end if
        localrsII.close
                localsqlII = "select * from qx_roleprivil where roleid = '"
&username2ro
    leid(username)& "'and mdlid = '"& mdlname2mdlid(mdlname)&"'"
        localrsII.cursorlocation = 2
        localrsII.open localsqlII,conn,1,3
        if(localrsII.eof) then
            tmp2 = "0"
```

```
        else
            tmp2 = localrsII("privil")
        end if
        localrsII.close
        if tmp1>=tmp2 then
            name_mdlname2privil = tmp1
        elseif tmp1<tmp2 then
            name_mdlname2privil = tmp2
        end if
        set localrsII = nothing
    end function
    %>
```

程序首先根据用户名和模块名从用户权限表qx_userprivil中查询是否有赋给该用户此模块的访问权限，并将相应的值赋给一个临时变量temp1;然后程序根据用户名和模块名从角色权限表qx_roleprivil中查询赋给用户的角色中对模块相应的权限,并将该值赋给另一个临时变量temp2;程序最后比较temp1和temp2将用户的相应权限返回给Session对象。程序中用到两个自编函数username2 roleid(username)，作用是根据用户名从qx_userrole表中获取赋给用户的角色，mdlname2mdlid(mdlname)作用是根据模块名称从qx_module表中得到模块ID。

以五金管理模块为例，我们只要在每个页面的开始加上如下类似代码：

```
<SCRIPT LANGUAGE= " VBScript ">
Call IsLogin
if session("WJ_PRIVIL") < "1" then
    alert(" 您没有权限访问该模块 ")
end if
</SCRIPT>
```

就可有效地阻止没有权限的用户访问该页面，对于需要输入或修改数据之处只要判断sessin("WJ_PRIVIL")的值是否等于"2"就可以了。Call IsLogin的作用是为防止没有权限的用户，通过直接访问页面URL访问数据库。IsLogin函数用VBScript的具体实现如下：

```
<SCRIPT LANGUAGE= " VBScript ">
sub IsLogin
    if session("username") = "" then
        alert_go " 请先登录，然后再访问本系统 ","/login.htm"
    end if
end sub
</SCRIPT>
```

此外为防止某些人用诸如" abc'or'1'=1' "的表达式作为用户名和密码来避开对数据库的查询，强行登录系统。我们用了一个Javascript函数isansi()在用户按提交按钮的时候，判断表达式中是否包含'符号，进行预防处理。

```
</td>
<a><onclick="javascript:{if(isansi(txtusername.value)&&isansi
(txtpassword.value)
) submit();}">
<img  name="Image10" border="0" src="images/enter.gif" width="120"
height="31"
style="cursor:hand"></a>
</td>
```

3 结束语

ASP技术由于内置功能强大的组件，并且编程简单，正成为网络开发中的一种流行语言，ASP在实际系统开发应用中，由于它是在服务器端解释执行的，在客户端，它的源代码不能用浏览器查看到，安全性能比较高。本系统用ASP实现权限访问控制取得比较好的效果，当然系统的安全是多方面，权限访问控制还要和其他安全措施配合使用。

参考文献

1 林风、李维章、赵莉，动态网站设计捷径 ASP [M]，西安电子科技大学出版社，1999。

2 David Austin著，周生炳译，Oracle 8使用指南，电子工业出版社，1999。