基于角色的表单验证安全的研究

孙仁鹏

（南京信息职业技术学院 软件学院，江苏 南京 210046）

摘 要：为了在 ASP.NET 中进行灵活的权限管理，提出了一种自定义的表单验证机制，即利用已有的安全验证机制，将用户的角色内嵌入票据中，手动发出票据，通过 Global.asax、Web.config 和 HTTP Module 的使用判断用户的权限来完成，同时使用 2 的幂次方保存用户的角色集合信息，方便了角色的处理。

关键词：表单验证；URL 授权；HTTP 模块；角色；票据；2 的幂次方

中图分类号：TP309 **文献标识码**：A

Research of role-based forms authentication security

SUN Ren Peng

(Nanjing College of Information Technology, Nanjing 210046, China)

Abstract: In order to ASP.NET in the flexibility of rights management, a custom forms authentication mechanisms, namely the use of existing security authentication mechanism to the user's roles within the embedded ticket, the manual notes issued by Global. asax, Web.config and the use of Http Module to determine the user's permission to complete, while the use of 2-th power mean to preserve the user's role in the collection of information, facilitates the handling of the role.

Key words: forms authentication; URL authorization; HTTP Modules; role; ticket; 2-th power mean

Web 应用程序的安全性是必需的，也越来越重要。在过去，只向已授权用户提供页面，通常通过人工在每个页面中嵌入服务器端代码的方式实现，繁琐、不灵活且效率低。在 ASP.NET 中表单验证并不直接支持基于角色的验证，本文剖析了如何利用.NET 的验证和授权机制，将安全架构代码从页面中移动到 HTTP 请求管道中来，并构建了支持基于角色授权的表单身份验证，满足了实际的应用需求。

1 ASP.NET 管道

当 HTTP 请求进入 ASP.NET Runtime 以后，它的管道由托管模块（Managed Modules）和处理程序（Handlers）组成，并且由管道来处理这个 HTTP 请求，其中 Modules 可以进行请求前的预处理和响应后的再处理，Handlers 处理实际的请求和响应，如图 1 所示[1]。

ASP.NET 中身份验证和授权机制是使用 HTTP 模块对象实现的，HTTP 模块对象是作为标准 ASP.NET 管道处理的一部分进行调用的，各个 Web 请求和响应都通过对象管道进行传递。

模块能在实际处理程序执行前后进行过滤，所以是添加安全代码的理想位置。当请求和响应通过管道传递时，模块中的相应事件会在某些处理阶段触发。

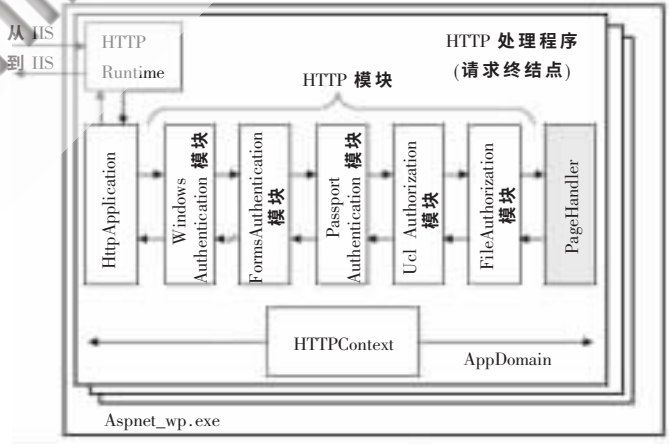

图 1 ASP.NET 管道处理

HttpApplication 对象控制管道处理，其中订阅了这些事件，各个 HTTP 模块与这些事件挂钩，这样，可编写应用程序事件处理代码，分离应用程序逻辑和安全逻辑，自动执行安全代码的调用。

下面介绍表单验证所需的主要模块：

FormsAuthenticationModule：在 AuthenticateRequest 事件中，检查表单认证票据来认证用户，将未授权的用户导航到登录页面。表单认证票据一般存放在用户的 cookies collection 里的，如果没有表单认证票据，那么这个用户就是匿名的。

UrlAuthorizationModule：在 AuthorizeRequest 事件中，判断是否授权当前用户访问所请求的 URL，该模块需要根据应用程序的配置文件指定的授权规则来执行。

2 表单验证流程

表单验证是一个基于票据的系统。URL 授权是 ASP.NET 通过在 Web.config 文件中设置单独的网页、文件目录和 Web 服务等授权规则，由 URL 授权模块来执行[2]。图 2 阐述了表单验证流程。

匿名用户访问受保护资源时，表单验证模块会把用户重定向到登录页面，原始请求的 URL 被保存为参数，以便以后使用。

如果成功登录，就可以生成身份验证票据，写入 Cookie，发送到客户机（该票据以后随 HTTP 请求自动传递），随后由表单认证模块重定向最初请求的 URL。

再次请求原始 URL 时，表单验证模块会从 Cookie 中获取身份验证票据，URL 授权模块会利用用户信息来判断该用户是否有权访问受保护的资源。

客户端有可能不支持 Cookie，在这种情况下，ASP.NET 也可以使用无 Cookie 的表单认证票据。在这种模式下，将表单认证票据编码进行 URL。

3 在电子商务中具体实现

3.1 使用 2 的幂次方表示角色

角色可以用枚举类型表示，定义如下：

```
[Flags]
    [Serializable()]
    public enum Enum_AccountRight : long
    {
        StateNull = 0,
        总台呼叫 = 1,
        商品销售 = 2,
        订单管理 = 4,
        商品采购 = 8,
        客户服务 = 16,
        商品信息管理 = 32,
        供应商管理 = 64,
        客户信息管理 = 128,
        系统设置 = 256,…
    }
```

特性[Flag]表示可以用“|”、“&”等操作符组合枚举值，其中“&”操作符可检查是否存在某个角色，“|”可用于角色分配。数据库中存储的用户角色为角色集合子集的和，提高了数据的安全性。

3.2 权限表设计

SysAccount 表的结构如表 1 所示。

3.3 Web.config 配置

3.3.1 配置表单验证

必须在 Web.config 文件中正确配置表单验证，其中 <authentication mode = "forms"> 表示应用程序采用表单验证方式，loginUrl 指定如果没有找到任何有效的身份验证 Cookie，未登录将请求重定向到的 URL，defaultUrl 用来指定登录成功后默认转向的页面，基本配置如下：

```
<authentication mode = "Forms">
    <forms defaultUrl = "Pages/NavigationPage.aspx"
loginUrl = "Pages/LoginManager/LoginWeb.aspx"timeout="30">
    </forms>
</authentication>
```

3.3.2 配置授权规则

通过配置 URL 授权，

图 2 表单验证流程

表 1 SysAccount 表的结构

字段名	Account ID	Account Name	Employee ID	Password	AccountRight
注释	系统自动分配的帐号	登录帐号名	员工号	密码	角色集合，长整形

拒绝未通过身份验证的用户访问相应资源。以下配置说明只有角色为“系统设置”的用户才能访问 AccountManager 目录下的资源，也可以配置对指定文件的访问权限，基本配置如下：

```
<location path="Pages/AccountManager">
  <system.web>
    <authorization>
      <allow roles="系统设置"/>
      <deny users="*"/>
    </authorization>
  </system.web>
</location>
```

URL 授权模块提供了以 user-by-user 或 role-by-role 原则来限制对一系列页面的访问，更具维护性的办法是使用基于角色的授权规则。

3.4 在验证票中缓存角色，手动发出验证票

构建验证票，在验证票中使用 userData 属性存储角色，手动发出验证票。然后为每个请求从票据中获取角色，避免了反复访问数据库获取角色，同时还可以方便加密。具体实现流程如图 3 所示。

(1)二进制匹配使用 CAST 函数；

(2)Cookie 超时值的读取使用 AuthenticationSection 类或使用 XmlDocument 解析 web.config 文件；

(3)验证票据的生成使用 FormsAuthenticationTicket 类，加密票据使用 FormsAuthentication 类的 Encrypt 方法（为 Web 应用程序管理 Forms 身份验证服务）；

(4)不支持 Cookie 的判断使用 FormsAuthentication 类的 CookiesSupported 属性，票据的传递使用 SetAuthCookie 方法；

(5)Cookie 的生成使用 HttpCookie，送入客户端使用 Reponse 类；

(6)重定向初始请求页使用 FormsAuthentication 类。

3.5 自定义基于角色的表单身份验证

ASP.NET 授权 HttpApplication 对象负责处理所有的请求，并且提供方法和事件，以便能够编写可以使用 HttpApplication 事件的自定义程序。

将角色和用户关联，使用基于角色的授权，方法是通过在 AuthorizeRequest 事件中使用角色和用户标识（Identity）构建主体（Principal）对象。这是非常重要的，因为下游授权模块使用此 Principal 对象来确定是否授权。

Identity 对象代表当前用户所需的基本信息，借助于 Identity 对象，应用程序可以找到属于当前线程的用户，

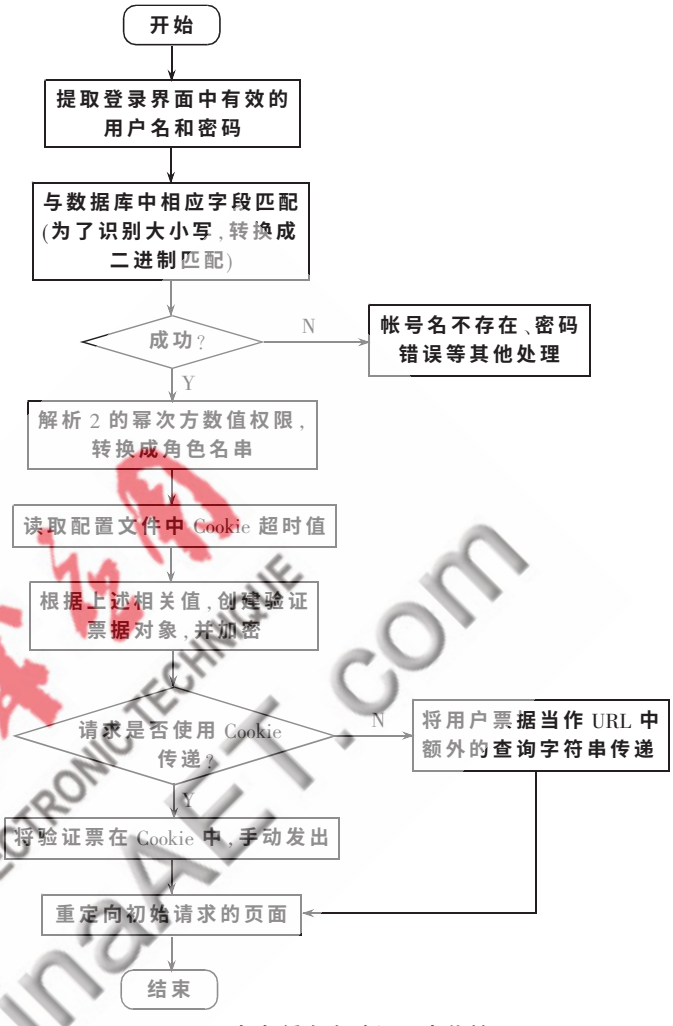

图 3 角色缓存在验证票中传输

Principal 代表用户当前安全上下文，包括用户标识和当前用户保存的信息（如角色）等组成。

每个请求都将调用 AuthenticateRequest 事件处理程序，之后由 UrlAuthorizationModule 对应用程序定义的角色自动进行角色检查，通过调用 Context.User.IsInRole 以确定是否为授权用户来完成的，因此在授权模块运行前设置 Context.User，具体实现流程如图 4 所示。

开发自定义角色的窗体身份验证方案有两种：

(1) 利用 Global.asax 文件。只需在 Application_AuthenticateRequest 事件体中实现如图 4 所示流程。

(2)使用自定义的 HTTP 模块。继承 IHttpModule 接口实现方法，然后在 web.config 中注册自己。

IHttpModule 接口定义了两个方法：

①Init：允许 HttpModule 注册对应于 HttpApplication 对象中的事件的处理方法。

②Dispose:用于释放 HttpModule 中占用的系统资源。

所以只要 HttpApplication 对象引发相应事件，都将调用 HttpModule 中方法。

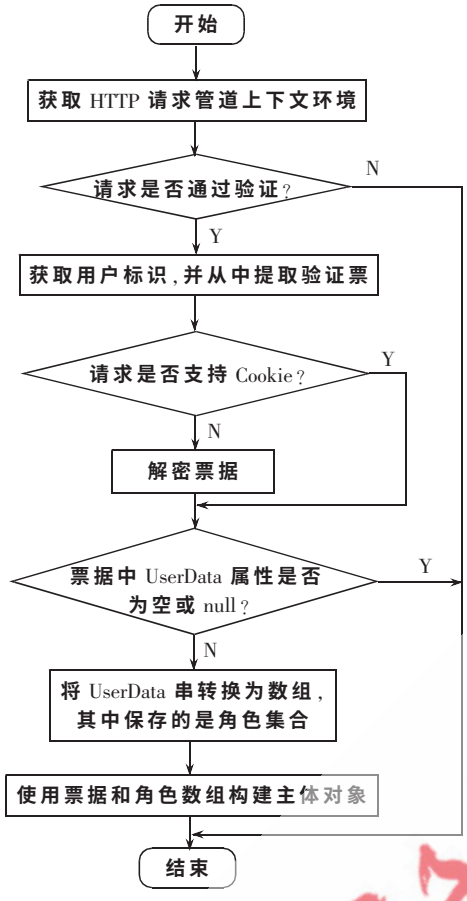

图 4 自定义基于角色的表单验证身份

3.6 员工角色维护

在后台管理中，利用 GridView 控件来完成员工角色的显示、编辑和删除。整个过程需要在模板中使用 CheckBoxList 控件显示枚举类型中所有的角色名称。显示时使 SysAccount 表中员工长整形表示的角色复选框被选中，更新时取 CheckBoxList 控件中被选中的角色名对应的枚举组合值，插入到对应员工的帐号表中，维护过程需将角色名称和对应的值做转换。

3.7 保护表单验证

(1) 使用安全套接字层 SSL(Secure Sockets Layer)保护从浏览器传递到服务器初始登录的用户名或密码。

(2) 不使用持久性 Cookie，对敏感的视图状态进行散列或加密。

(3) 对票据运用安全措施，将<forms>元素里 protection 设为 all，可防止别人读取和修改票内容。

(4) 在访问用户存储区时使用参数化的存储过程，防止 SQL 注入攻击。

(5) 对敏感数据进行加密，比如数据库中用户的密码，Web.config 中数据库连接字符串。

3.8 代码中授权检查

前面的描述只能控制页面或文件夹的访问，那么如何控制访问页面中特定的功能呢？这就需要在运行阶段检查角色或者用户，将用户没有权限的超链接和按钮失效或隐藏。

本文深入分析了 HTTP 请求管道，利用了 ASP.NET 提供的验证和授权机制，采用其扩展性来自定义表单验证行为，便于权限设计的扩展和维护。本文提供了一个正确、明智和快速的方式完成基于角色的表单验证机制，方便了资源和功能的保护，以代替在系统中使用复杂的 ASP.NET 成员资格 API。当然，通过本文的论述，可以更好地理解成员资格 API。该权限设计在南京苏泽电子商务系统中得到了较好的应用。

参考文献

[1] Microsoft TechNet.安全指南[CP/OL].http://www.microsoft.com/china/technet/security/Guidance/secmod37.mspx.

[2] SCOTT M. An overview of forms authentication[CP/OL]. http://www.asp.net/learn/security/tutorial-02-cs.aspx.2009.

[3] 张子阳.ASP.NET 架构[CP/OL].http://www.cnblogs.com/JimmyZhang/archive/2007/09/04/880967.html.2009.

[4] 王涛,杨季文.ASP.NET WebForms 底层请求处理机制初探[J].计算机应用与软件,2007,24(10):120-121.

[5] BARIER D.开发更安全的 ASP.NET 2.0 应用程序[M].北京:人民邮电出版社,2008.

[6] M.M.AL-Farooque Shubho.Forms authentication and role based authorization[CP/OL].(2009-05-30).http://www.codeproject.com/KB/web-security/RolesFormsAuthorization.aspx.

[7] Heath Stewart.Role-based v security with forms authentication[CP/OL]. http://www.codeproject.com/KB/web-security/formsroleauth.aspx.

(收稿日期:2009-12-29)

作者简介:

孙仁鹏,男,1972 年生,硕士,主要研究方向:Web 技术,分布式技术和数据库技术。