

# 嵌入式 Web 访问控制系统的设计与实现

谯 倩,毛燕琴,沈苏彬

(南京邮电大学 计算机学院,江苏 南京 210003)

**摘 要:**针对嵌入式 Web 系统自身的安全,结合嵌入式 Web 系统的特点,在对基于角色的访问控制模型研究的基础上对其进行简化修改,去掉角色继承的复杂模式,在此基础上提出了适用于嵌入式 Web 系统的“用户-角色-权限集(业务-页面-操作)”访问控制设计方案。并利用 CGI 技术实现了特定的嵌入式 Web 应用系统的访问控制功能,限制了合法用户对嵌入式 Web 系统资源的访问,防止了非法用户的侵入或因合法用户的不慎操作而造成的破坏。对实现的 Web 应用系统进行了测试,测试结果表明该模型具有良好的功能。

**关键词:**CGI; 嵌入式 Web; 访问控制

**中图分类号:**TP302.1

**文献标识码:**A

**文章编号:**1673-629X(2011)08-0228-05

## Design and Realization of Embedded Web Access Control System

QIAO Qian, MAO Yan-qin, SHEN Su-bin

(School of Computer Science and Technology, Nanjing University of Posts  
and Telecommunications, Nanjing 210003, China)

**Abstract:** For the security of embedded Web system itself, combined with the characteristics of embedded Web system and based on the research on the model, it simplifies RBAC model to remove the role of complex patterns of inheritance and gives the embedded Web solution for access control system that is “user-role-privilege set (business-page-operation)” model. The embedded Web access control system is achieved through CGI technology, limiting user access to embedded Web systems resources, and preventing the intrusion of unauthorized users or the damage caused by careless operation of legitimate users. The Web application system was tested, and the test results show that the model has good functions.

**Key words:** CGI; embedded Web; access control

## 0 引 言

随着嵌入式系统网络化发展,嵌入式 Web 系统以其独特的优势得到了广泛应用。把 Web 用于智能家居以及其它嵌入式系统中,使用户能够通过浏览器对远程设备进行监控,为用户进行操作提供了很大方便。目前应用比较广的 Web 开发方法有:CGI、ASP、JSP、PHP 等,但由于嵌入式设备资源有限等原因,CGI 以它独特的优势和为绝大多数服务器支持的特点成为嵌入式 Web 服务器应用程序开发的首选<sup>[1]</sup>。

嵌入式 Web 系统在提供经济、实用的接入方案的同时,其本身的安全也成为了一个重要课题。因此必

须建立一套有效的安全访问控制机制,来保护嵌入式 Web 系统资源。文中在对访问控制模型研究的基础上,对 RBAC 模型进行简化,并结合倪晚成等人提出的“基于角色-页面模型”,给出了适用于嵌入式 Web 系统的“用户-角色-权限集(业务-页面-操作)”访问控制设计方案。并通过 CGI 技术实现了对特定 Web 系统中的访问控制,从而保证系统自身的安全。

## 1 相关技术

### 1.1 CGI 技术

CGI(Common Gateway Interface)定义了 Web 服务器与 CGI 脚本之间的接口标准。其主要功能是在 Web 环境下,从客户端传送一些信息给 Web 服务器,Web 服务器把接收到的有关信息放入环境变量,然后再去启动所指定的 CGI 脚本即外部扩展应用程序以完成特定的工作,CGI 脚本从环境变量中获取相关信息来运行,最后以 HTML 格式输出相应的执行结果返回给

收稿日期:2010-12-21;修回日期:2011-04-02

基金项目:国家高技术(863)计划项目(2006AA01Z208);江苏省科技支撑计划项目(BE2009157);南邮青蓝计划项目(NY208023)

作者简介:谯 倩(1985-),女,山东齐河人,硕士研究生,研究方向为计算机网络;沈苏彬,研究员,博士研究生导师,研究领域为计算机网络、下一代电信网、网络安全、嵌入式软件、网络计算。

浏览器端。按照 CGI 标准编写的外部扩展应用程序可以处理来自客户端的协同工作数据完成客户端与服务器的动态交互,从而实现静态 HTML 网页无法实现的功能,如表单数据处理、数据库查询等。由于用户能传递不同的参数给 CGI 脚本,所以 CGI 技术使得浏览器和服务器之间具有交互性<sup>[2]</sup>,但其程序必须运行在服务器端。

CGI 本质上是一种服务机制<sup>[3]</sup>,它是在 Web 服务器上定义了 Web 客户请求与应答的一种方法。当客户端向 Web 服务器发出 HTTP 请求,客户向服务器的请求只要属于 CGI 范围,服务器收到请求后就启动 Web 服务器的一个 CGI 程序。该 CGI 程序可以调用其它外部程序,以完成前台的指令。当系统完成任务之后,再把执行结果传给 HTTP 服务器。它的任务是把客户的请求从环境变量和标准输入 Stdin 中取出并进行相应的加工处理,待处理结束后由 CGI 程序通过标准输出决定如何对客户的请求做出应答,并将处理结果以 HTML 格式回送到客户端浏览器。

通用网关接口 CGI<sup>[4]</sup>是 Web 服务器和外部程序的一个接口。CGI 可以使编写的程序处理 WWW 上客户端送来的表单和数据并对此做出某种反应。在服务器中,HTML 实现客户与服务器端的静态信息交互;CGI 程序则满足客户和服务器的动态交互要求。CGI 的工作分为如下几个步骤:

(1) 客户端使用 TCP/IP 协议,与服务器建立连接,向服务器端发出访问请求。

(2) Web 服务器收到请求,判断请求的内容,如果是 CGI 程序,则激活相应的 CGI 程序。而 CGI 程序所需要的参数则通过环境变量获取。

(3) 执行 CGI 程序,对客户端的请求做出反应,同时 CGI 程序也可以调用相应的外部程序来完成操作。

(4) Web 服务器根据 CGI 程序的处理结果,对 CGI 的输出按照一定的格式传送给客户端。

(5) 客户端浏览器将 CGI 程序的输出显示在浏览器的窗体上。

(6) Web 服务器中断和客户端的连接。

更明确地说,CGI 仅是在 Web 服务器上可执行的程序代码,而它的工作就是按照控制信息要求产生并传回所需的文件。由此可见 CGI 程序在用户和服务器之间进行交互查询时起着重要作用。

## 1.2 访问控制技术

访问控制的目的是限制访问主体对关键资源及系统资源的访问,防止非法用户的侵入或因合法用户的不慎操作而造成的破坏<sup>[5]</sup>。

访问控制模型主要有:自主访问控制模型 DAC、强制访问控制模型 MAC、基于角色的访问控制模型

RBAC 等。其中 RBAC 模型是目前应用最广泛的一种模型,甚至被认为是 DAC 和 MAC 的最佳替代者<sup>[6]</sup>。

2000 年 David F. Ferraiolo 等人共同提出了一个 RBAC 标准<sup>[7]</sup>,该标准除了统一常用术语外,还提出了一个 RBAC 参考模型,见图 1。

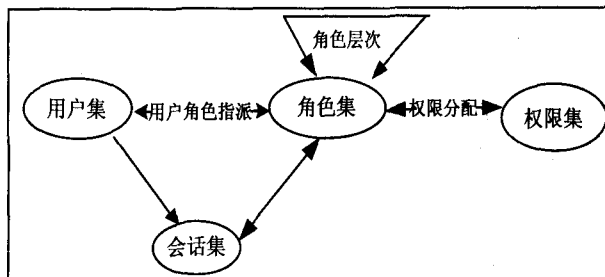


图 1 RBAC 访问控制模型

RBAC 的核心思想<sup>[8]</sup>是:将访问权限和角色相联系,通过给用户分配合适的角色,让用户与访问权限相联系,用户可以在角色间进行转换,系统可以添加、删除和修改角色,还可以对角色的权限进行添加、删除和修改。这样通过应用 RBAC 将安全性放在一个接近组织结构的自然层面上进行管理,通过对角色的授权来控制用户对系统资源(客体)的访问。

在此模型中,用户是系统中可以访问资源的主体;角色是授权的集合,指一个组织或者任务中工作或位置,代表一种权利和责任<sup>[9]</sup>;一个用户可以有一个或多个角色。权限是指用户对系统资源进行操作访问的许可。角色作为中间桥梁将权限与用户联系起来,并将用户和权限的分配分成用户角色指派和角色权限指派;通过角色权限指派使权限与角色关联,一个角色可以拥有多个权限。当某个角色拥有了某些权限后,就可以进行这些权限所允许的操作,而这个角色所对应的用户也将拥有这个权限。会话是一个动态概念,用户激活角色时就会建立会话。在 RBAC 模型中,对资源访问许可通过不同的角色实现,通过给用户分配合适的角色,使用户与访问权限相联系进而间接地访问资源,所以可以实现用户与权限的逻辑分离,进而提高权限管理的效率与灵活性。

嵌入式 Web 系统是一些 Web 网页和用来完成某些任务的其它资源的集合。它以界面友好、方便用户操作、支持无限用户、系统扩展性强和易维护等优势成为现代信息系统的主流。B/S 架构下嵌入式 Web 系统中,页面是系统的主要表现形式,处理好对页面的访问控制是解决嵌入式 Web 系统访问控制问题的关键。因此对嵌入式 Web 系统的访问控制可变为对用户所能访问的 Web 页面和内部数据的控制。

对 Web 系统,倪晚成等人提出了“基于角色-页面模型”<sup>[10]</sup>的 Web 用户访问控制方法。但在这个模型中把页面作为权限分配的基本单位,一个页面实现的

功能对应一个权限,从而可以通过控制页面对用户的可见性来控制用户访问。这种方式使得权限的划分依赖于页面功能,从而不能灵活地对权限进行控制。如果对权限的控制力度太细,则会造成页面太多,如果对权限的控制力度太粗,则对用户的访问管理缺少灵活性。

## 2 嵌入式 Web 访问控制的设计

文中通过对 RBAC 模型和“基于角色-页面模型”的研究,在二者的基础上提出“用户-角色-权限集(业务-页面-操作)”的访问控制机制,来限制和控制外界对嵌入式系统资源的访问能力。该模型将 RBAC 模型进行简化去除了角色继承的复杂关系。同时结合 Web 系统的特点,将 RBAC 模型中的权限集进行划分,形成业务-页面-功能模式。这种模式不再以页面为单位,而是以业务为单位,还可以根据具体需求灵活地决定是否对业务进行细分,弥补了“基于角色-页面模型”中的权限的划分仅依赖于页面功能的不足。该模型如图 2 所示。

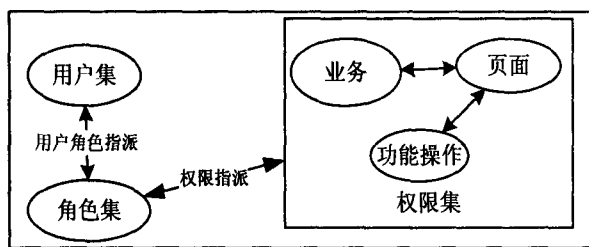


图 2 访问控制模型

用户是系统中可以访问资源的主体,系统中可以对其访问的资源即为客体。对资源访问许可被封装在不同的角色中,一个用户可以从属于一个或多个角色,角色与权限关联,一个角色可以拥有多个权限。当某个角色被指派了某个权限时,它就可以进行这个权限所允许的操作,而这个角色对应的所有用户也将拥有这个权限。在此模型中,用户被指派到角色,用户通过角色间接地访问系统资源。在权限集中,一个业务可能由一个或多个页面组成,而每个页面又可能由一个或多个小的功能操作构成,这就使得角色的划分不是以页面为单位,而是以业务为单位,并且根据系统需求灵活地决定是否对业务进行细分。

图 2 所提出的模型,体现了 RBAC 的核心思想,将访问权限与角色相联系,用户和权限之间通过角色关联。用户和角色之间的指派关系及用户的添加删除可通过一个用户管理子模块实现;角色和权限集之间的指派关系及角色的添加删除修改可以通过角色管理子模块实现,其中角色对应访问策略的制定即是对角色的权限进行添加、删除和修改操作;而对系统的资源的

访问可以通过访问控制子模块实现。在权限集中,一个业务由一个或多个页面组成,而每个页面又由一个或多个小的功能操作构成,因而对嵌入式 Web 系统资源的访问控制可分为两个部分。首先对页面的访问即权限集中的业务-页面,可利用 CGI 技术,根据用户角色生成对应的系统菜单,过滤掉用户无权访问的业务模块,只显示用户有权限访问的业务菜单,进而决定可访问哪些页面。其次对页面中子功能的访问即页面-操作,则是对其细粒度控制的一个体现,当进入某个页面时先根据用户信息判断用户是否是已登录系统以此防止未登录系统的使用者进行强行操作,然后根据用户角色及所对应的权限进行判断可防止已登录用户通过直接输入 URL 地址等非法方式来访问页面中的某些未授权功能。

根据以上对访问控制子模块的分析设计可从以下三个方面实现对系统的访问控制:

(1) 用户身份验证。采用基于知识的身份验证<sup>[11]</sup>,即利用 CGI 技术实现对用户名、密码的验证。它是对系统进行访问控制的基础。

(2) 对系统业务的访问:基于访问者的角色进行验证。根据用户角色,利用 CGI 脚本动态生成系统菜单,以此控制用户所能访问的业务模块。

(3) 对页面的访问。采用在客户端设置 cookie 及服务器端设置 session 信息的机制实现<sup>[12]</sup>。系统为每个成功登录的用户创建一个 session,该 session 中包含对系统进行访问控制所要用的信息,比如该用户的角色,登录时间等等。用户要访问系统页面或者页面中某一项具体操作时,首先根据自身 session 信息此用户的角色获取相应的访问策略,并判断是否具有访问权限。结合 CGI 技术进行控制,防止未授权用户通过直接输入 URL 地址等非法方式来访问未授权的页面或者已授权页面中的某些未授权功能。

本系统中用户登录并访问系统的整体流程框架如图 3 所示。

用户登录的过程即为身份验证的过程,这是对系统进行访问控制的基础,在此过程判断用户是否为该系统用户。如果用户有权登录系统,则转入访问控制模块,通过身份验证后,进入系统首页,首先根据自行设置的 session 信息判断用户角色,用 CGI 技术生成动态菜单,显示用户有权访问的业务模块。其次访问系统资源时,通过菜单访问某一页面时同样根据设置的 session 信息对用户进行验证,判断用户角色,显示相应的功能,防止用户直接输入 URL 就可访问页面。

## 3 嵌入式 Web 访问控制的实现

根据图 2 所示的访问控制模型,功能实现可分成

三个子模块:用户管理子模块、角色管理子模块、访问控制子模块。

用户管理子模块完成用户的添加、删除、用户密码的修改及用户角色的分配及转换。

角色管理子模块完成角色的添加、删除及对应的访问策略配置等。

访问控制子模块则是从纵向角度使用户根据用户角色对应的访问策略访问系统,如图3所示。

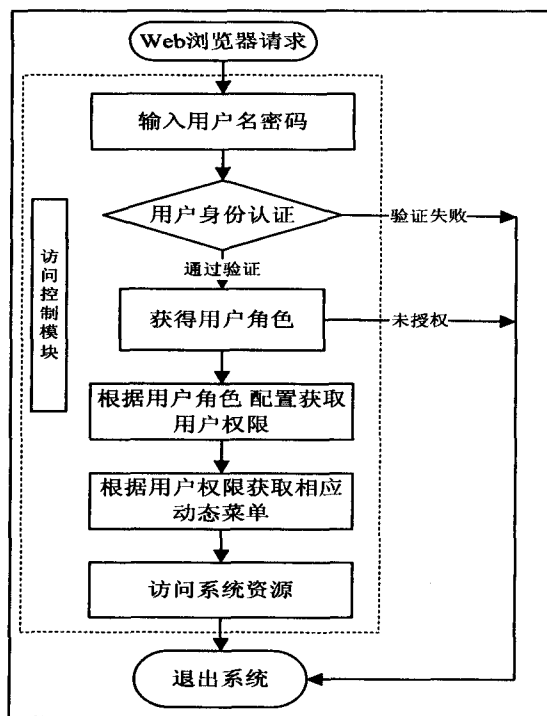


图3 系统纵向访问流程图

以上各模块均是通过 CGI 技术具体实现,其中访问控制子模块通过在服务器端设置 session 信息实现。首先在用户成功登录系统时在服务器端创建 session 信息,此信息中包含用于访问控制的信息;其次当用户进入系统进行访问时,先要获取此 session 中用户的角色等对访问控制有用的信息并据此进行相应操作,来确定用户能访问哪些系统资源。

信息的创建包含客户端 cookie 信息和服务器端 session 信息的创建。cookie 信息存在于客户端浏览器中,在用户访问系统时就创建,用户通过此信息获取服务器端的 session 信息。session 信息在用户登录成功的同时创建,并作为一个文件存储在服务器端,以登录时的用户名后加一个随机字符串为此文件名。

●服务器端 session 信息内容包含以下几个方面:

- (1) 用户的最后一次活动时间。用来检查用户是否长时间没有操作,视为已经退出登陆。
- (2) 一个随机的字符串 id。用来验证客户端的身份,这个字符串同时作为 cookie 发往客户端。
- (3) 实际要存储的数据。例如用户名、对应角色

等。根据此部分的信息判断用户是否有权访问系统某些资源。

●客户端 cookie 信息设置实现代码如下:

```

void set_cookie()
{
    time_t t;
    long int sess_id=time(&t);
    sprintf(str_now,"%ld",sess_id);    //id
    char username[10];
    cgiFormStringNoNewlines("username",username,
10);//获取用户名
    char see[40];
    sprintf(see,"%ld,%s\n",sess_id,username);
    printf("Set-Cookie:%s\n",see);
}
  
```

通过 printf("Set-Cookie:%s\n",see) 来设置 cookie。

当访问系统资源时首先要获取用于访问控制的信息,再根据此信息进行访问控制授权。以后在访问每个页面时都要根据客户端浏览器中的 cookie 信息进行操作。

(1) 根据浏览器中的信息检查是否登录超时,若超时则进入登录界面重新登录。

(2) 根据 cookie 中的用户名,获取登录用户在服务器端的 session 信息包含用户的角色,登录时的 id 等。

(3) 对比 cookie 中的 id 和 session 文件中的 id,验证客户身份,未通过验证则需重新登录。

(4) 根据用户角色对用户授权,觉得用户有权访问哪些资源。

(5) 刷新服务器端 session 中最后活动时间,以便进行超时检测。

●B/S 模式下客户端登录并访问系统过程:

(1) 客户端通过 IE 浏览器输入 ip 地址,登录网络服务器。进入登录界面输入登录信息,点击登录。

(2) 调用 CGI 程序获取 session 信息对用户进行验证并通过 CGI 技术,根据其角色生成相应的动态菜单,显示用户有权访问的功能模块。当通过菜单访问某一页面时同样根据设置的 session 信息获取用户的角色,显示相应的功能,防止用户直接输入 URL 就可访问页面。

(3) CGI 将执行结果发送网页到电脑客户端的 IE 浏览器。

定义角色 A 对应的用户 ad1 和 ad2,角色 B 对应的用户 user。其中角色 A 对应管理员角色,B 对应普通用户角色。两个角色的初始配置策略为:A 角色拥

有系统所有业务功能的访问权限,而 B 角色仅对系统中一部分资源具有访问权限。两个角色的访问权限可参见图 4、图 5 所示菜单。

●从以下两个方面测试系统:

(1)不同角色用户登录系统。即分别为用户指定不同的角色登录系统并进行操作。

测试结果如下:

图 4、图 5 分别为不同角色对应的用户 ad1、user 登录系统后的状态。两个用户的角色不同,在系统具体实现时表现为如图所示,两者生成的动态菜单也不同。

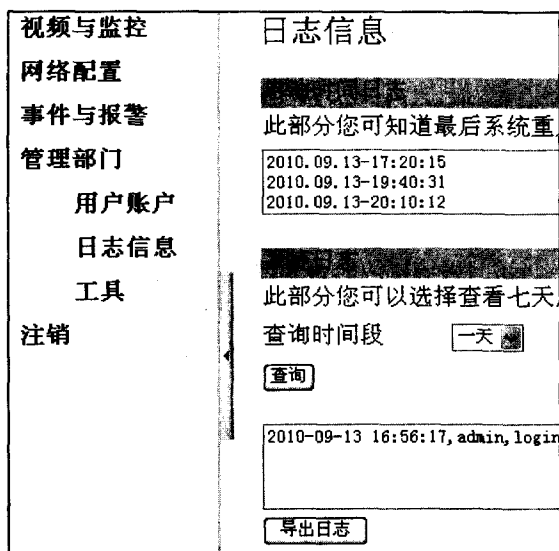


图 4 ad1 用户登录后显示的日志信息界面

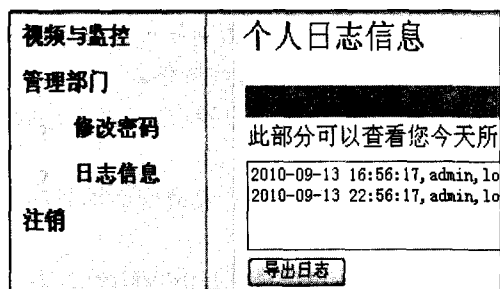


图 5 user 用户登录后显示的日志信息界面

(2)同一角色对应的不同用户,修改此角色所对应的访问策略,观察该角色策略修改前后各用户的系统登陆状态,以此测试访问策略配置方法,即角色权限的指派关系。

重新配置角色 A 的访问策略,方便起见,角色 A 重新配置后的策略与角色 B 对应的策略相同。配置前角色 A 对应的用户 ad1 和 ad2 登录系统后的状态相同,如图 4 所示,配置后 ad1 和 ad2 登录状态也相同,如图 5 所示。针对同一角色的用户 ad1, ad2, 重新配置其角色所对应的访问策略后,观察前后两次登录系统

的返回结果即对比图 4 和图 5,证明该角色所对应的所有用户在角色的访问策略发生变化后均受到了影响。

## 4 结束语

文中是在嵌入式 Web 系统开发基础上展开,通过对访问控制技术的研究,结合“基于角色-页面模型”中的不足,对 RBAC 模型进行修改,给出了对嵌入式 Web 系统的“用户-角色-权限集(业务-页面-操作)”的访问控制模型,并最终利用 CGI 技术在嵌入式 Web 系统中实现此方案,限制了用户对嵌入式系统资源的访问,防止了非法用户的侵入,保证了嵌入式 Web 系统自身的安全性。

## 参考文献:

- [1] Wang Zhenxing, Ren Xianyi, A Study on Cgi of Embedded Web server[C]//2008 International Symposium on Computer Science and Computational Technology. [s. l.]: [s. n.], 2008:480-483.
- [2] 张曦璜,柴志雷. 嵌入式 WEB 服务器中 CGI 的特点及实现[J]. 小型微型计算机系统, 2003, 24(11):2046-2048.
- [3] 吕 峥,朱逸芬. CGI 程序与 FORM 表单交互的实现[J]. 计算机应用, 1999, 19(3):24-26.
- [4] Chen Tianhuang, Huang Jiaxi. Design and Realization of CGI in Embedded Dynamic Web Technology[C]//Network and Parallel Computing Workshops, 2007. IFIP International Conference. [s. l.]: [s. n.], 2007:774-777.
- [5] 刘宏月,范九伦,马建峰. 访问控制技术研究进展[J]. 小型微型计算机系统, 2004, 25(3):56-59.
- [6] 乔 颖,须 德,戴国忠. 一种基于角色访问控制(RBAC)的新模型及其实现机制[J]. 计算机研究与发展, 2000(1):37-44.
- [7] Ferraiolo D F, Sandhu'R, Gavrila S, et al. Proposed NIST standard for role-based access control[J]. ACM Trans on Information and System Security, 2001, 4(3):224-274.
- [8] 李孟珂,余祥宣. 基于角色的访问控制技术及应用[J]. 计算机应用研究, 2000, 17(10):44-47.
- [9] 黄益民,杨子江,平玲娣,等. 安全管理系统中基于角色访问控制的实施方法[J]. 浙江大学学报, 2004(4):408-413.
- [10] 倪晚成,刘连臣,刘 伟. 基于角色-页面模型的 Web 用户访问控制方法[J]. 计算机工程与应用, 2006, 42(21):124-126.
- [11] 沈苏彬. 网络安全原理与应用[M]. 北京:人民邮电出版社, 2005.
- [12] 周若谷,李宗伯. 嵌入式环境下 CGI 程序的 Session 实现方案[J]. 电脑知识与技术, 2008, 2(10):81-99.