基于监听和过滤器技术的Web服务身份验证

龙鹏飞,乔 波
(长沙理工大学 计算机通信与工程学院,湖南 长沙 410076)

摘 要:为了保证Internet上Web应用的安全,防止信息被非法访问,文中提出了一种基于监听和过滤器技术的Web服务身份验证方法,分析了其工作原理和核心技巧,并阐述了J2EE平台上的监听和过滤器技术,为在企业级Web应用开发中运用监听和过滤器技术提供了一定的指导作用。经过测试,文中提出的Web服务身份验证方法,能够在用户发出请求信息时,先对用户的请求进行监听,再对用户的请求进行过滤,从而有效地阻止非法访问而导致的信息泄露,保证存储在Web服务器上的页面文件的安全。

关键词:Web服务;监听;过滤器;身份验证
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1005-3751(2006)03-0135-02

Sniffer and Filter-Based Web Services Identity Validation

LONG Peng-fei, QIAO Bo
(Computer & Communication Eng. Coll., Changsha University of Sci. & Tech., Changsha 410076, China)

Abstract: In order to assure security of Web applications on Internet and avoid illegal accesses, a method for Web services identity validation based on sniffer and filter is proposed in this paper. Its principle and key techniques are analyzed. Sniffer and filter technologies in the J2EE platform are expounded. It can be concluded that the presented method can guide the application of sniffer and filter in enterprise Web application development. Testing results show the requests can be sniffed first, and then filtered by using the presented method when user's requests are sent out. So, the information is effectively prevented from leaking due to illegal accesses, and the security of page files stored into Web server are also kept.

Key words: Web services; sniffer; filter; identity validation

0 引 言

随着电子商务的迅速崛起,基于Web的应用模式迅速发展。Web应用从局部化到全球化,从B2C(business-to-customer)发展到B2B(business-to-business);从集中式发展到分布式。Web服务成为电子商务的有效解决方案。当今国内外占统治地位的Web服务器,如Apache,IIS(Internet Information Server)等,对于存储在其中的信息缺乏足够的安全措施(一般都是明文),仅仅将它们置于防火墙、IDS、以及操作系统的保护之下,一旦被黑客攻破,即会造成重要信息的泄露,带来严重后果。为了保证Internet上Web应用的安全,防止信息被非法访问和修改,需要采用安全控制或信息加密等手段。现有的安全技术如数字签名、XML加密标准、访问控制技术等,一定程度上解决了特定的安全问题[1~3]。文中提出了一种基于监听和过滤器技术的Web服务身份验证方法,运用J2EE中的监听和过滤器技术可以对用户进行访问验证,能有效地增强Web服务的安全性。

收稿日期:2005-06-17
作者简介:龙鹏飞(1960—),男,湖南长沙人,教授级高级工程师,硕士研究生导师,研究方向为计算机软件研究与开发。

1 J2EE及Filter技术

J2EE是Sun公司定义的一个开发分布式企业应用的规范。它提供了一个多层次的分布式应用模型和一系列开发技术规范。多层次分布式应用模型是指把一个完整应用的功能分解到多个相对独立的服务器和组件,组件运行在服务器提供的容器内,容器为组件提供各种服务,组件间通过一些协议来实现相互通信。J2EE技术规范可以为软件的安全性、可扩展性、可复用性提供强有力的保证。

过滤器技术是Servlet 2.3规范中的重要功能,过滤器是一个类似于由容器管理的Servlet对象,能以声明的方式插入到HTTP请求响应过程。其主要作用是在客户端请求到达被请求的服务之前,或者服务响应离开服务器到达客户端之前,根据需要对请求或响应进行预处理。

采用Filter技术后,客户端和服务器端之间的一切交互都可以采用Filter来实现管理,如图1所示。Web应用中的过滤器截取从客户端进来的请求,并做出处理的答复。它可以说是外部进入网站的第一道关,在这个关卡里,可以验证客户是否来自可信的网络;可以对客户提交的数据进行重新编码;可以从系统里获得配置的信息;可以过滤掉客户的某些不应出现的词汇;可以验证客户是否已经登陆等等。

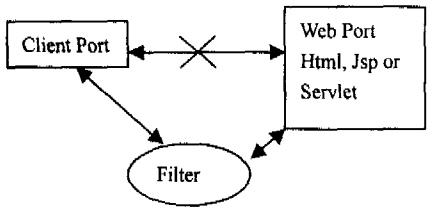

图 1 Filter 功能图

可以为一个 Web 应用组件部署多个过滤器，这些过滤器组成一个过滤链，每个过滤器只执行某个特定的操作或者检查，这样请求在达到被访问的目标之前，需要经过这个过滤链，如果由于安全的问题不能访问目标资源，那么过滤器就可以把客户端的请求拦截，如图 2 所示。

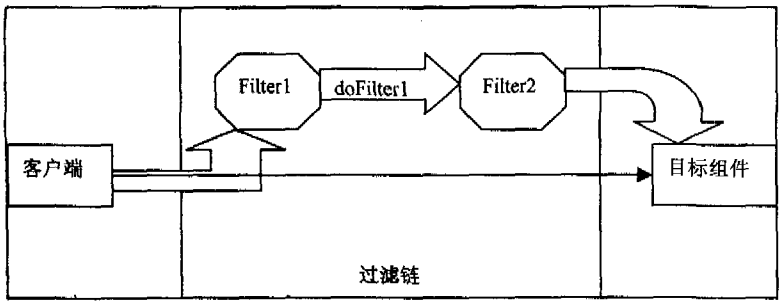

图 2 过滤器链图

2 Web 监听及其设计

在 Servlet 2.4 规范中，新增加了一个技术，就是可以监听客户端的请求。一旦能够在监听程序中获得客户端的请求，就可以对请求进行统一处理。下面将设计一个 Web 管理程序，可以监听请求信息，如果在本机访问，就可以不登录；如果是远程访问，就必须要登陆：要实现对客户端请求的和请求中参数设置的监听，需要实现 ServletRequestListener 和 ServletRequestAttributeListener 接口。在实现接口的程序中有一个 requestInitialized 方法，在这个方法里，它将获得客户端请求对象，然后通过这个请求对象来获得访问的客户端 IP 地址，如果这个地址以“127”开始，那么认为它是从本机访问的，就在请求中设置一个 isLogin 的属性，并把这个属性值设置为 Boolean(true)对象，如果不是从本机访问，就把这个属性设置成 Boolean(false)对象。

3 Filter 的设计与实现

过滤器类必须实现接口 Javax. Servlet. Filter。Filter 接口包括以下 3 个方法[4]：

(1)init()方法，完成过滤器初始化，通过一个 FilterConfig 对象把配置信息发送到相应的过滤器；

(2)doFilter()方法，完成过滤行为的方法，也是过滤器中前驱过滤器调用的方法，该方法引入的对象 FilterChain 使得前驱过滤器调用 doFilter()方法的同时调用下一个过滤器；

(3)Destroy()方法，在容器关闭或应用程序实施撤消时完成过滤器退出服务。

下面是一种 Filter 接口的实现，用于对用户的访问验证完成，如果验证通过，那么允许访问指定的资源；如果验证不通过，那么把请求转发到登陆的界面。

部分代码如下：

```
//过滤处理的方法
Public void doFilter(final ServletRequest req, final ServletResponse res, FilterChain chain)throws IOException, ServletException
{
  HttpServletRequest hreq=(HttpServletRequest)req;
  HttpServletResponse hres=(HttpServletResponse)res;
  HttpSession session=hreq.getSession();
        String isLogin="";
        Try{
        isLogin = (String) session. getAttribute("isLogin");
          if(isLogin. equals("true"))
          {
          System. out. println("在 SignonFilter 中验证通过");
          //验证成功,继续处理
          Chain. doFilter(req. res);
    }
    else
    {
    //验证不成功,让用户登录
    hres. sendRedirect(LOGIN_PAGE);
    System. out. println("被 SignonFilter 拦截一个未认证的请求");
    }
   }
  ......
}
```

在上述程序的 doFilter 方法中，首先通过 isLogin=(String)session. getAttribute("isLogin")代码来获得用户是否登陆的属性，如果这个属性是“true”，表示用户已经登陆，那么就可以访问目标资源；如果这个属性是“false”，表示用户没有登陆，那么把请求转发到登陆页面。

另外企业级应用程序的开发、配置与部署也是重要的部分。从部署角度看，过滤器是整个 Web 应用程序的一部分，与 Servlet，JSP 或静态资源属于同一个层次。用标准的 J2EE 应用程序部署配置过滤器，可以分为定义和映像两个层次。过滤器定义用于编译器通报与过滤器相关的文本名称。在 web. xml 中定义 Filter 时，需要指定 Filter 的名字、Filter 的实现类，如果有参数，那么要配置它的参数。而过滤器映像主要是做一些 URL 的映射，它和 Filter 映射匹配的请求会被处理。上面例程的 Filter 的配置如下：

……

(下转第 172 页)

务器控件的 operator 属性设置成:DataTypeCheck,限制输入数据的格式应符合日期类型的数据,论文字数的 Validation 服务器控件的 Type 设置成 Integer, MinimumValue =n1, MaximumValue=n2,限制输入仅仅为数字,并且在 n1,n2 范围之内(n1,n2 是一般一篇论文的字数)。

表单提交后,本系统通过编写的类进行进一步检查,在这个类中包括所有的需要检查的输入。该类中没有数据成员,只由一些检查函数组成,使用时定义对象,通过调用对象的方法来检测输入的合法性。

3 系统的特点

(1)使用了当前的最新技术和主流技术。该系统设计是基于 B/S 和 C/S 模式相结合的方式实现的,而且使用 .NET 框架,实现了跨平台。在.NET 框架中,程序代码被编译成微软中间语言(Microsoft Intermediate Language, MISL),当要运行程序时,运行阶段语言(Common Language Runtime,CLR)接管,进一步将 MISL 代码编译成计算机的本机语言。进而可以运行到任何类型的计算机上。

(2)系统充分利用校园网络的数据资源。本系统中科研人员的相关资料及人员的统计都直接来自学校的人事档案数据库,这样就减少了该系统的数据库输入量,又作到了数据的统一。

(3)实现多种语言文字的录入。本系统可以录入中、英、日、韩、俄等多种文字。论文可能发表在各国的期刊上,有多种文字形式,本系统采用 ASP.NET 与 C# 实现。涉及到写入、修改数据库内容时,用存储过程完成,这样不仅提高了系统的运行速度,还可以实现多种文字的并存,而直接使用 SQL 语句时则无法达到这一目的。

4 结束语

本系统采用最新技术.NET 框架实现了基于 C/S、B/S 结构相结合的科研管理系统。充分利用了校园网络资源,既方便用户的使用,为广大用户提供熟悉的 Web 界面(B/S 模式实现),又便于科研部门的数据统计,保证科研部门的日常业务处理(C/S 模式实现)。

目前,高校科研管理系统已在我校投入使用,并取得良好的效果,获得了一致的好评。对不足之处将做进一步完善,使系统更加安全稳定。

参考文献:

[1] 陈 旭. BWS 模式及其应用研究[J]. 计算机应用研究,2001,18(6):32-34.

[2] Ferracchiati F C.. NET 数据服务 C# 高级编程[M]. 毛尧飞译. 北京:清华大学出版社,2002.

[3] Payne C. ASP. NET 入门到精通[M]. 赵 斌,等译. 北京:人民邮电出版社,2002.

[4] 廖信彦. ASP. net 交互式 Web 数据库程序设计[M]. 北京:中国铁道出版社,2003.

[5] 童德利,田 娟,谢 琪. 基于 B/S 模式的构件式酒店管理信息系统的设计与实现[J]. 计算机应用研究,2003(4):126-129.

[6] 唐家才. SQL Server 2000 中文版管理实务[M]. 北京:人民邮电出版社,2001.

[7] Sceppa D. ADO. NET 技术内幕[M]. 梁 超译. 北京:清华大学出版社,2003.

(上接第 136 页)

```
<web-app>
  <filter>
    <filter-name>SignonFilter</filter-name>
    <filter-class> SignonFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>SignonFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

在 Filter 的映射中,SignonFilter 的一个 URL Pattern 为"/*",这表示对任何 URL 请求都会被过滤,这样在请求到达目标组件之前,会经过 SignonFilter 的处理,如果在 SignonFilter 中验证通过,那么将允许访问目标组件;如果认证不通过,那么请求就被拦截[5]。

4 结 论

文中提出的 Web 服务身份验证方法,能够首先对客户端的请求进行监听,然后对客户端的请求进行过滤,从而有效地阻止了黑客的非法访问而导致的信息泄露,保证了存储在 Web 服务器上的页面文件的安全。但应当注意,Web 服务器的安全是一项系统工程,必须全面实施才能确保安全。只有将过滤技术与防火墙、VPN、SSL 以及灾难应急恢复等其它安全技术融合在一起,才能全方位地保障 Web 服务器的安全。

参考文献:

[1] 岳 昆,王晓玲,周傲英. Web 服务核心支撑技术:研究综述[J]. 软件学报,2004,15(3):428-442.

[2] 杨 涛,刘锦德. Web Services 技术综述——一种面向服务的分布式计算模式[J]. 计算机应用,2004,24(8):1-4.

[3] 张建华,李 涛,刘晓洁,等. Web 页面加密存储及访问机制[J]. 计算机工程,2004,30(13):97-98.

[4] 李 松,沈文轩. J2EE 平台上过滤器技术的研究与应用[J]. 鞍山科技大学学报,2004,27(3):190-193.

[5] 飞思科技产品研发中心. JSP 应用开发详解[M]. 北京:电子工业出版社,2005.