基于 AJAX 和 SAML 技术的互联网单点登录系统

A Single Sign – on System for Internet Website Based on Ajax and SAML

唐四薪　邹 赛　谢新华　（衡阳师范学院 计算机科学系 湖南衡阳 421008）

摘　要: 传统的互联网网站认证模式要求用户在每一家网站上分别注册,以后才能登录该网站,单点登录技术 SSO 是实现集成身份认证和统一授权服务的有效方法,通过对传统 SSO 分析,提出了一种适用于互联网的基于 AJAX 和 SAML 技术的统一单点登录系统,并在网络货币系统项目中用代码实现。

关键词: Ajax SAML 单点登录

1　引言

由于不同的门户网站、用户社区网站和商业服务网站采用不同的登录机制,各网站通过设置独立的用户注册登录系统,使用户在每一家网站注册均需填写一份个人信息资料,用户被迫保持多个身份,从而导致孤立的业务关系和用户体验。对于一个经常上网的人来说,可能需要在几百家网站注册,每天有大量的时间浪费在重复注册和登录的时间中,因此,设计一个只需注册一次,就可以在互联网上任何网站通过认证的单点登录系统(Single sign – on,SSO)是我们的开发目标,为此消除这种访问孤立的关键是建立一种联合的身份认证。

传统的互联网多次登录的缺点在于:多次登录使用户身份安全性下降;每个服务单位自带登录系统软件和保存用户身份认证的数据库,使服务系统过于庞大,且造成重复建设,加大了投资成本;多次登录程序繁琐,浪费用户时间,给用户使用造成不便;大多数网站登录采用用户名和口令字的方法,没有密码学意义[1]。

单点登录系统就是为解决传统登录验证机制所存在的问题而出现的,所谓单点登录就是指用户只需在网络中主动地进行一次身份认证,之后无需另外验证身份,便可以访问其被授权的所有网络资源[2]。但目前的单点登录产品,一般是通过对用户的集中管理,来实现单点登录功能的,这就造成了一个 Web 服务对用户的信任不能被网络中的其它服务共享。而互联网单点登录系统则能一次登录整个网络,它不但可以通过减少登录次数提高用户的上网效率,而且在电子商务或电子政务中,单点登录由于可向任何信任站点提供用户的身份,金钱等资料,使用户能在所有这些网站中更方便的进行认证,签名,购物等活动。

2　单点登录系统的相关技术

2.1 AJAX 技术及优点

AJAX 全称为" Asynchronous JavaScript and XML"(异步 JavaScript 和 XML),是指一种创建交互式网页应用的网页开发技术。Ajax 不是一种新技术,而是由几种蓬勃发展的技术以新的强大方式组合而成,它包括:使用 XHTML 和 CSS 实现标准化呈现;使用文档对象模型 DOM(Document Object Model)实现动态显示和交互;使用 XML 和 XSLT 实现数据交换和操作;使用 XMLHttpRequest 实现异步数据检索;最后用 JavaScript 将所有这些绑定到一起[3]。

传统的 Web 网站强制用户进入提交/等待/重新显示范例。用户总是在请求后的等待响应之中度过。而 Ajax 提供与服务器异步通信的能力,从而使用户从请求/响应的循环中解脱出来[4]。一般的 web 应用允许用户填写表单(form),当提交表单时就向 web 服务器发送一个请求。服务器接收并处理传来的表单,然后返回一个新的网页。这种做法浪费了许多带宽,因为在前后两个页面中的大部分 HTML 代码往往是相同的。由于每次应用的交互都需要向服务器发送请求,

应用的响应时间就依赖于服务器的响应时间。这导致用户界面的响应比本地应用慢得多。与此不同,AJAX 应用可以仅向服务器发送并取回必需的数据,它使用 SOAP 或其它一些基于 XML 的 web service 接口,并在客户端采用 JavaScript 处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少,结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成,所以 Web 服务器的处理时间也减少了。

2.2 基于 Ajax 的 SAML 单点登录技术

由于越来越多的系统通过 Web 服务、门户和集成化应用程序彼此链接,对于保证欲共享的信息安全交换的标准的需求也随之日益显著起来。SAML(安全性断言标记语言,Security Assertion Markup Language)是一个基于 XML 的、用来交换安全信息的框架,而安全信息是关于主体(Subject)的一组断言(Assertions)[5]。其中一个关键概念是身份联邦,它可满足 SAML 的定义,也就是说可使用独立、受管理的多个信息来源中的信息,从而实现身份验证这样的安全服务。由于 SAML 是基于 XML 的,在提交票据和生成登录 Session 的过程中可以与同样基于 XML 的 Ajax 技术相结合,因此基于 Ajax 的 SAML 技术可减少受信网站 WEB 服务器的工作量。其具体流程如图 1 所示:

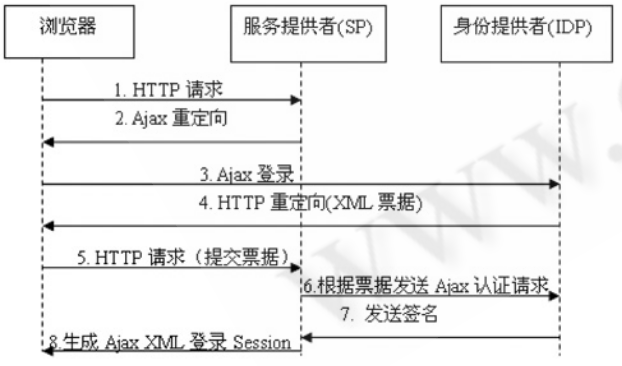

图 1 基于 Ajax 和 SAML 的单点登录原理图

基于 Ajax 的登录时序流程采用 Browser/Artifact 方式[6]:(1)用户访问 SP;(2)SP 检查自身的用户 Session,如果 Session 存在,表明用户已登录,直接跳转到(8),否则重定向到 IDP 的登录界面;(3)IDP 提示用户登录;(4)如果验证成功,IDP 即生成用户的身份认证断言和 Artifact,并建立与 Artifact 的对应关系,然后将此 Artifact 作为参数向用户发送 HTTP 重定向指令;(5)用户重定向到 SP;(6)SP 根据此 Artifact 向 IDP 发送 Authentication Request 请求;(7)IDP 查询 Artifact 与断言的对应表后,将签名发送给 SP;(8)SP 收到断言,如果断言表示验证成功,则生成用户登录 Session,并将已登录的 Ajax 无刷新界面返回给用户,用户登录成功。

3 互联网单点登录系统设计

统一单点登录系统采用 B-S 结构,整个系统由一个认证服务器和应用系统服务器(该系统用于实验目的,因此这里的应用服务器只有一个,在实际应用中应该是互联网上所有带有登录系统的网站)及嵌入 Ajax 代码的用户客户端构成,其中认证服务器负责对用户进行身份验证,并把认证结果返回给应用服务器,应用服务器根据认证的结果,结合其自身的 RBAC 访问控制系统[7],对用户访问系统进行授权。图 2 是系统结构设计:

用户上网只要注册登录过认证服务器的认证网页,用户的登录信息就被记录在服务器的数据库中,当用户再浏览其他网站的网页时,其他网站就会根据用户 Session 识别用户是否使用了统一网站单点登录系统,如何判断用户最近一段时间内使用了系统中某个服务,是系统安全的重要问题,Session 管理模块就是用来解决此问题的,根据该模块功能,可将 Session 管理模块分成 Session 授权(Session Authority)和 Session 容器(Session Recipient),共同对用户的 Session 信息进行管理。每个 Web 服务器上都会部署一个 Session 容器,而 Session 授权被部署在一台服务器上。用户必须首先得到 Session 授权,并在 Session 授权上创建一个新的 Session。当用户使用某个 Web 服务时,必须得到部署在该服务器上的 Session 容器的允许。认证服务器根据用户的授权,将一定限度的用户信息传给这些网站,(如对于普通网站,认证服务器只要将用户名信息传给网站就可以了,而对于论坛等网站,认证服务器需要将用户的 id,等级,所拥有的论坛积分等信息传给这些网站),这些网站由此识别出用户身份。

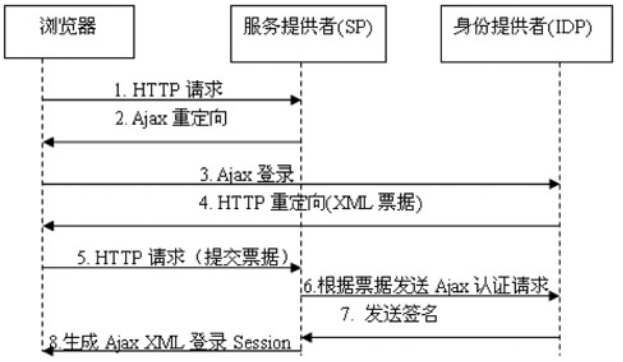

图 2　互联网单点登录系统结构设计

4　系统的核心程序和实现

4.1 用户客户端的关键代码

AJAX 引擎发出 XMLHTTPRequest 请求的时候，不再是指向某个静态地址，而是将其请求的 URL 地址指向服务器端声明的映射到 XMLAssemblyFactoryServlet 的地址，并且 AJAX 引擎在浏览器端向服务器端的 Servlet 发出请求时可以携带多个参数来限定所需要的数据和操作。

```
var url = "XMLAssemblyFactoryServlet?paraName1 = para – Valuel&p2 = v2";
xmlHttpReq. open("GET",url,true) '发送请求
xmlHtttpReq. onreadystatechange callback;
xmlHttpReq. send(null);
```

4.2 认证服务器的关键代码

根据认证服务器端的票据（Artifact），构造 SAML 请求对象，并把查询断言传递到认证服务器，认证服务器根据其保存的断言信息，返回查询结果（SAML 返回对象）给该页面，应用系统端根据该结果决定是否给用户访问系统的权限。当用户选择了需要进入的应用系统（网站）并确认进入后，系统将根据用户 ID 对应于该应用系统的用户帐户产生断言和票据，并把它保存在全局 Application 变量里。代码如下：

```
Dim assertion As Assertion '创建断言
=
CreateAssertion (ConfirmationMethod. Methods. Artifact. userCode)
Dim artifact As BrowserArtifactProfile. ArtifactType1 = CreateArtifact()
'创建票据
Dim artifactString As String = artifact. ToString()
'建立断言和票据的关联并存入全局变量 Application
Application. Add (artifactString, assertion)
```

生成断言后，系统将把票据以 URL 传值的方式传到应用系统端，代码如下：

```
stringBullder. AppendFormat (http://localhost/SAMLServiceProvider2/SAML/ArtifactReceive. aspx?TARGET = {0} &SAMLart = {1}", HttpUtility. UrlEncode (target), HttpUtility. UrlEncode (artifactString))
Response. Redirect (stringBullder. ToString (), False)
```

登录请求生成模块为用户构造一个针对目标系统的登录请求，这个信息中的内容包括目标系统 URL、用户身份信息，请求生成时间、请求有效时间、随机数以及认证服务器对上述信息的签名信息。

4.3 应用系统服务器的关键代码

首先在 web. xml 中声明 XMLAssemblyFactoryServlet 和它的 URL 映射。在 XMLAssemblyFactoryServlet 中采用 doPost 方法设置响应 Content Type 为" text/xml"，在完成一系列数据存取及业务逻辑后，将所有的有效数据包装在正确的 XML 格式当中，最后通过 HTTP 响应发回到客户端。其中的有效数据指对 SAML 请求进行断言，断言过程的实质是利用安全认证中心的私钥对 SAML 响应进行数字签名的过程。SAML 响应的主要内容就是 SAML 断言信息。

```
Response. setContentType(" text/xml")
Response. setHeader(" Cache – Control"," no – cache");
If (null ! = request. getAttribute(" Name1")) {'解析参数
paraValue1 = request. getAttribute(" Name1"). toString();
} … 解析其它参数
PrintWriter out = response. getWriter();
… 在此调用 JDBC，EJB 等进行数据存取，计算及业务处理
String resultXMLContent = " < matrix > < status >
```

SAML Information < /status > < /matrix > " ；

´将 XML 格式的 SAML 断言信息写到 response 中

Out. println(resultXMLContent)；

至此，通过 XMLAssemblyFactoryServlet 作为桥梁，用户客户端嵌入的 Ajax 技术与服务器端的 SAML 技术结合起来，使系统达到更加完美的效果，由于单点登录系统中传递的是纯粹的数据，对用户而言，系统显得比普通的应用更快了，单页面的操作也更为友好。

5 结束语

互联网单点登录系统的声明采用 SAML 描述，而 SAML 继承了 XML 跨平台的优点，使系统克服了以往单点登录系统受平台限制的制约，并适用于浏览器或客户端方式，系统消息之间的保密由 WS – Security 保证，满足了 Web 服务环境下端到端的安全级别要求。通过向互联网上各类网站进行推广，就能实现 Internet 一次登录，通行于所有网站身份验证的构想，这将极大地促进电子商务和电子政务的发展，因此具有很好的应用前景。

参考文献

1 Maler E, Philpott R. Assertions and Protocols for the OASIS Security Assertion Markup Language(SAML). http://www. OASIS – opera. org/committes/ documents. php. wg – abbrev = secutity, 2003 – 03.

2 陈小云. 统一身份认证系统的研究与实现. 成都: 西南交通大学, 2007.

3 Garrett J J. Ajax: A New Approach to web Applications. http://www. javalohhv. ora/articles/ajax (Accessed Feb. 2, 2005)

4 余翔宇. Ajax 技术及其框架实现. 软件技术研究, 2006,(9): 29—30.

5 Manish Verma. Ensure portable trust with SAML. http://www – 128. ibm. com/developerworks/xml/ library/ x – seclay4/index. html.

6 Thomas. Security analysis of the SAML single sign – on browser/artifact profile. Computer Security Applications Conference. 2003: 298 – 307.

7 Sandhu R, Coyne E J, Lfeinstein H L. Role – based access control models. IEEE Computer, 1996, 29(2): 38 – 47.