基于.NET Passport 的 B2C 网站研究与实现

Research and Implementation of B2C Web Based on .NET Passport

沈士根 （嘉兴学院 信息工程学院 314001）

摘要：单一登录.NET Passport 协议正逐渐成为 B2C 网站身份认证的关键技术。文章分析了.NET Passport 的工作原理，提出了基于.NET Passport 的 B2C 网站逻辑性设计，说明了在开发基于.NET Passport 的 B2C 网站时涉及的关键技术，最后结合授权管理器，给出了具体实现。

关键词：单一登录 .NET Passport 身份认证 B2C

1 引言

.NET Passport 作为一套完整的 Internet 身份认证管理系统，它可以让 Internet 用户只需要使用一组登录帐号和密码，即可登录所有加盟.NET Passport 的网站，即实现了单一登录 SSI（Single Sign – In）服务。.NET Passport SSI 将原先的身份认证由各个站点独立完成改为集中式认证、分布式授权，这使得网站管理者不必再亲自管理身份认证问题。

2 .NET Passport 工作原理

整个.NET Passport 由.NET Passport 帐户数据库和不同功能的网络服务器（如 Registration Server，Member Service Server，Update Server，Login Server 等）组成。其中帐户数据库储存了注册用户的帐户信息和加盟站点的 Site ID 及加密密钥；Registration Server、Member Service Server 和 Update Server 负责处理用户帐号的建立和修改。已向.NET Passport 登记加盟的 B2C 网站可使用 .NET Passport 的 Login Server，通过 SSI 服务完成用户的认证。在实现时，.NET Passport 依赖于 HTTP 重定向和 Cookies 之间的结合。

2.1 注册用户的帐户信息

一个.NET Passport 用户帐户包含用户的唯一识别码 PUID（Passport Unique Identifier）、用户凭证数据、用户配置数据等。PUID 在用户建立帐号时由.NET Passport 系统创建，其长度为 64 位，用于唯一标识.NET Passport 用户帐号。用户凭证数据包含用户的电子邮件地址（作为登录的帐户名）和密码（至少 6 位）。用户使用凭证数据登录.NET Passport，若帐户名和密码验证成功，则将该帐号的 PUID 加密发送到加盟.NET Passport 的 Web 站点。注意此时发送的是帐户的 PUID 而非帐户信息，从而保证了帐户信息的安全。用户配置数据包括个人姓氏、名字、性别、出生年月、所在国家等信息，用户可以指定配置数据是否公开给.NET Passport 加盟站点。

2.2 身份认证过程

当用户登录一个加盟.NET Passport 的 Web 站点时，用户、加盟 Web 站点和.NET Passport 之间会发生较多的交互作用。

（1）用户单击加盟 Web 站点上的“登录”按钮。

（2）加盟 Web 站点将用户的浏览器通过 HTTP 重定向到 www.passport.com 的一个特定页面，同时，在这个特定页面的 URL 后以 Query String 参数形式附加了加盟站点的 Site ID（站点加盟.NET Passport 时，.NET Passport 向站点颁发一个唯一的 Site ID）。

（3）如果 Site ID 有效，.NET Passport 将用户浏览器重定向到 login.passport.net 的用户登录界面。

（4）用户输入凭证数据（包括电子邮件地址和密码）。

（5）一旦用户提交信息后，.NET Passport 将用户输入的凭证以 SSL 方式发送到 www.passport.com 域。注意此时用户凭证仅送到 www.passport.com，其他加盟 Web 站点均不会收到此验证数据。

（6）.NET Passport 将用户帐号（电子邮件地址）和密码与.NET Passport 数据库中的记录作验

证。若验证成功，www. passport. com 从数据库中取出用户的 PUID 和用户公开的配置信息在用户端建立一个名为 MSPSec 的 Cookie，保存本次登录的信息，MSPSec 包括经 SSL 加密的用户 PUID。

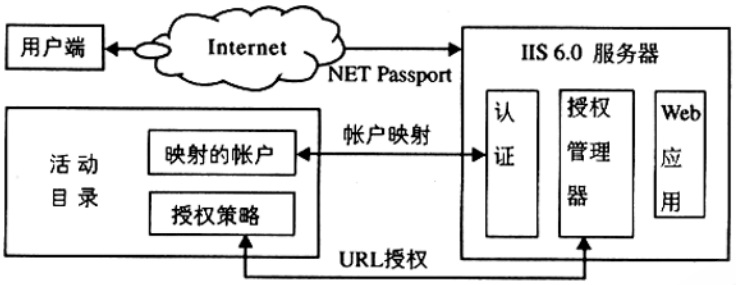

图 1　基于. NET Passport 的 B2C 网站逻辑设计图

(7) www. passport. com 用加盟站点的加密密钥(该密钥为对称密钥，在站点加盟. NET Passport 时由. NET Passport 创建)通过 3DES 加密算法加密用户票据(ticket)和配置(profile)信息，将它们作为加盟站点 URL 的 Query String 参数，使用户浏览器重定向到加了参数的加盟站点 URL。这时的用户票据包含了用户的 PUID 和 timestamp(默认值为 4 小时，表示 Cookie 的有效期)。

(8) 用户返回到加盟站点后，加盟站点上的 Passport Manager 对象获取 Query String 参数值，再使用解密密钥(与(7)中的加密密钥相同)解密用户票据和用户配置信息，利用解密用户票据后得到的 PUID 和 timestamp 在加盟站点域名下建立一个名为 MSPAuth 的 Cookie；利用解密用户配置信息后得到的用户配置信息在加盟站点域名下建立一个名为 MSProf 的 Cookie。此后，加盟站点可根据用户的 PUID 和用户配置信息为用户提供个性化服务。

(9) 当用户浏览其他加盟. NET Passport 的站点时，并不一定要重新登录验证的动作。因为. NET Passport 会以用户端的 Cookies 来决定用户是否登录、是否要更新 Cookies 数据，然后使用其他加盟站点的加密密钥加密用户票据和用户配置，再传送到其他的加盟站点。此过程实现了真正的单一登录，即一旦登录，就可浏览所有加盟. NET Passport 的站点。

(10) 当用户单击"退出"按钮时可以执行注销。

3　基于. NET Passport 的 B2C 网站逻辑设计

图 1 展示了基于. NET Passport 的 B2C 网站逻辑设计图。

(1) 当用户存取加盟. NET Passport 的 B2C 网站时，. NET Passport 提示用户提供合法的. NET Passport 帐号。

(2) 用户输入帐号后，经. NET Passport 认证，若认证成功，IIS 6.0 映射用户的 PUID 到活动目录(Active Directory)中的映射帐户。

(3) IIS 6.0 根据用户的凭证从授权管理器(Authorization Manager)请求对 Web 应用程序的授权。

(4) 如果授权管理器鉴定用户是活动目录中合适组(如"Trial User"组，该组在建网站时建立，属于该组的用户可被授权访问 Web 应用程序)的成员，授权管理器再将授权用户能否访问的信息返回 IIS 6.0。在实现时，授权管理器通过存放在活动目录中的授权策略来管理具体的授权。

(5) IIS 6.0 根据授权用户能否访问的信息返回用户请求的网页。

4　B2C 网站开发关键技术

4.1　判断用户是否已登录

当用户存取. NET Passport 加盟的站点网页时，经常要判断该用户是否已登录. NET Passport。在开发时，可先实例化 Passport Manager 对象，再调用 Passport Manager 对象的 IsAuthenticated 方法，根据该方法的返回结果(True 表示已登录)，而显示不同网页内容。

4.2　单一登录

如果用户未登录. NET Passport，或需要重新登录. NET Passport 时就需将用户浏览器重定向到. NET Passport 的登录服务器。在实现时，有三种方式：

(1) 调用 Passport Manager 对象的 Loginuser 方法直接将用户浏览器重定向到登录服务器的登录页面，用户经过认证后返回到加盟站点网页。然后可调用 IsAuthenticated 方法来确认用户是否已登录。

(2) 调用. NET Passport 对象的 LogoTag2 方法。如果用户未登录，该方法在登录页面显示"登录"按

钮,当单击该按钮后,再将用户浏览器重定向到登录服务器,如果登录成功,将显示"退出"按钮。

(3) 调用 Passport Manager 对象的 AuthURL2 方法。该方法类似于 LogoTag2 方法,但在登录页面显示的是超链接而不是按钮。

下面详细介绍单一登录的整个开发过程:

① 建立 Passport Manager 对象的一个实例 oMgr(该名自定)。

② 判断 oMgr. FromNetworkServer 属性值,若逻辑真,则表示用户刚从. NET Passport 被重定向到当前页面,此时必须清除带回的 Query String 参数值,可以通过再次重定向到不带 Query String 参数的当前页面的方法实现。

③ 调用 oMgr. IsAuthenticated 的方法。若返回逻辑真,表示用户有一个合法且未过期的. NET Passport 票据 Cookie;若返回逻辑假,调用 oMgr. LoginUser 方法将用户浏览器重定向到登录服务器,如果用户被认证,登录服务器将用户浏览器重定向到 LoginUser 方法中 returnURL 参数对应的地址。

④ 调用 oMgr. HexPUID 方法获取用户的 PUID。

⑤ 根据用户的 PUID 检查加盟网站的数据库(该数据库保存了用户是否同意使用他的配置数据的状态)。若用户已同意,则可调用 oMgr. Profile 属性获取用户的配置信息,如 oMgr. Profile("birthdate")可获取用户的出生年月。若用户未同意,将用户浏览器重定向到请求用户同意使用其配置信息的页面,若用户选择同意,则将该状态保存到加盟站点的数据库。

⑥ 调用 oMgr. LogoTag2 方法,因为此时用户已登录,所以在页面上显示的是"退出"按钮。

4.3 退出处理

(1) 用户单击任意加盟站点上的"退出"按钮后,用户浏览器被重定向到. NET Passport 的集中退出脚本(该脚本位于 www. passport. com,不需用户开发)。

(2) 集中退出脚本根据保存在客户端且处于 www. passport. com 域下,名为 MSPVis 的 Cookie 判定用户共访问了哪些加盟站点。

(3) 集中退出脚本删除所有 www. passport. com 域下的 Cookies。

(4) 集中退出脚本调用各加盟站点在. NET Passport 设置的 Expire Cookie URL 页面,该页面由用户开发,用于删除加盟站点域下所有与. NET Passport 相关的 Cookies。在开发时主要通过 Response. Cookies 删除名为 MSPAuth 的票据 Cookie 和名为 MSPProf 的配置数据 Cookie。如果登录时采用 SSL,则还应删除名为 MSPSecAuth 的 Cookie。最后返回 HTTP200 状态信息到集中退出脚本。

(5) 集中退出脚本根据各加盟站点返回的 HTTP200 状态,若用户成功退出,则在各登录过的站点列表旁打上绿色的"√",否则显示红色的"×"。

5 构建基于. NET Passport 的 B2C 网站

在具体构建时,Web 服务器使用 IIS 6.0 和 Windows Server 2003 操作系统,通过. NET Passport 实现认证过程,结合授权管理器实现授权过程。

5.1 配置与活动目录和授权管理器集成的. NET Passport

(1) 安装 Passport SDK。Passport SDK 是一套 COM 对象,只有安装 SDK 后,才可以在. NET 框架内使用. NET Passport 类。当调用. NET Passport 类的实例的方法时,实质是利用 COM 互操作调用由 SDK 安装的 COM 类的方法。SDK 包括对象模型和 Passport Manager 管理程序。本文涉及的是最新的. NET Passport 2.5 版本。

(2) 向. NET Passport 注册 B2C 网站。在 B2C 网站测试和部署. NET Passport 认证之前,必须向 Microsoft 注册以获得 Site ID 和加密密钥。获得 Site ID 和加密密钥的步骤包括:

① 登录 http://www. netservicesmanager. com,单击 Create and Manage an Application 链接。

② 选择测试/开发环境 Preproduction Application。还有一种环境类型为产品环境。当处于网站构建阶段时,应选择测试/开发环境。

③ 单击 Add Service,选择. NET Passport。

④ 在表单中输入的必填信息包括网站标题(Web Site Title)、域名(Domain Name)、默认返回 URL(Default Return URL)、保密策略 URL(Privacy Policy URL)、联合标志图像 URL(Co-brand Image URL)和 Cookie 期满 URL(Expire Cookie URL)。其中默认返回 URL 是指当用户单击"登录"按钮被重定向到.

NET Passport 登录服务器，再由登录服务器验证用户的凭据后要返回的 URL；保密策略 URL 能够通知站点中处于保密级别的用户；联合标志图像 URL 表示一个标志图像，它会在用户被重定向到. NET Passport 时与. NET Passport 图像联合出现；Cookie 期满 URL 用于删除 B2C 网站域名下所有与. NET Passport 相关的 Cookies。

⑤ 下载加密密钥安装程序 PartnerXXX. exe，其中 XXX 为网站的 Site ID。运行 PartnerXXX. exe /addkey 和 PartnerXXX. exe /makecurrent /t /0 安装加密密钥。

（3）在活动目录中创建一个试验组织单位 Trial Users（该名自定），在 Trial Users 组织单位下建立一个试验超级用户 Trialadmin（该名自定）和一个试验用户组 Trial User（该名自定）。设置 Trial User 的组作用域为本地域，组类型为安全组。设置 Trialadmin 对 Trial User 组完全控制的权限。设置 Trialadmin 委托控制 Trial Users 组织单位，并在安全性方面具有读和创建所有子对象的权限。

（4）运行管理工具下的 Active Directory 用户和计算机，在 Program Data 中创建一个容器类 Contoso（该名自定），设置 Trialadmin 委托控制 Contoso，并在安全性方面具有完全控制的权限。

（5）将 Trialadmin 添加为本地 Administrators 组成员。

（6）运行 IIS 管理器，在 Web 服务扩展中设置允许 Active Server Pages 和 ASP. NET V1. 1. 4322（或更新版本）。

（7）映射用户的 PUID 到活动目录的 Trialadmin。

（8）授权管理范围并分配角色。

5.2 配置使用授权管理器的 B2C 网站

要实现授权管理器的授权，必须使 IIS 6.0 能在授权管理器和授权范围对应的 URL 之间正常通信。

（1）创建一个使用本地系统帐户的应用程序池 LocalSysAppPool（该名自定）。

（2）分配 LocalSysAppPool 到 trialentry 虚拟目录。

（3）在 ContosoStore 的授权管理用户角色 Reader 下添加 B2C 网站的主机名。

（4）添加 c: \winnt \System32 \InetSrv 文件夹中的 URLAuth. dll 到 B2C 网站的通配符应用程序映射（执行顺序）区域。其中 c: \winnt 为操作系统的安装文件夹。URLAuth. dll 实质是具有 URL 授权 ISAPI 的拦截器。当每个用户请求 B2C 网站的 URL 时，用户请求首先进入 URLAuth. dll，然后，URLAuth. dll 再通过存放在活动目录中的授权策略决定对用户请求的授权存取。

（5）运行 IIS 管理器，在 Web 服务扩展中将 URLAuth. dll 添加为一个新的 Web 服务扩展并设置允许。

（6）可用记事本打开 c: \winnt \System32 \InetSrv 中的 metabase. xml（该文件存储了 IIS 的配置信息和策略信息），并设置相应的策略。

5.3 转换开发/测试环境到产品环境

在转换时，B2C 网站要同意. NET Passport 许可和服务协议，并登录 http://www. netservicesmanager. com，然后由 Microsoft 评定 B2C 网站是否遵循所有的. NET Passport 条例和 UI 向导，如果通过评定，那么 B2C 网站就可以从开发/测试环境转换到产品环境，从而提供实际的 B2C 服务。

6 结束语

通过结合使用 IIS 6.0 及 Windows Server 2003 中的授权管理器，使得用户访问 B2C 网站时先经过. NET Passport 的身份认证，再把对 B2C 网站网页的授权请求发送到授权管理器，由授权管理器根据制定的策略将授权决定传递给 IIS 6.0，若用户通过身份认证并得到授权即可访问 B2C 网站的相应网页。

参考文献

1 Microsoft. Microsoft . NET Passport Review Guide [EB/OL]. http://www. microsoft. com/net/services/passport/review_guide. asp, January 2004.

2 Microsoft. MSDN Library for Visual Studio . NET 2003 [M/CD], 2003.

3 Microsoft. Extranet Access Management [EB/OL]. http://www. microsoft. com/technet/security/topics/identity/idmanage/P3Extran _0. mspx, April 2004.