异构应用的单点登录方法①

郑　伟　王加阳　(中南大学 信息科学与工程学院 湖南 长沙 410083)

摘　要： 传统的 CAS 单点登录解决方案一般只针对门户网站等分布式 Web 应用，针对 C/S 与 B/S 模式并存且服务器端不允许修改的异构应用系统，并没有很好的解决方案。在对传统单点登录方案 CAS 研究的基础上，结合 B/S 与 C/S 模式的客户端对应的辅助技术即 Windows 消息机制及浏览器插件技术，设计并实现一种基于客户端辅助扩展和认证信息集中管理的单点登录解决方案。

关键词： 单点登录; 异构应用; 消息机制; 浏览器插件技术

Single Sign-On for Heterogeneous Applications

ZHENG Wei, WANG Jia-Yang

(College of Information Science & Engineering, Central South University, Changsha 230027, China)

Abstract: Traditional Single Sign-On solution is not suitable for the heterogeneous applications since it's generally for web portals and other distributed web applications, there isn't a suitable solution for the disparate applications in which both C/S and B/S mode exist, and whose server sides are not allowed to modify. This paper introduces the characteristic of the traditional single sign-on solution CAS, combines it with the assistant technology in B/S and C/S mode that are Windows message mechanism and browser plug-in technology, then proposes a new sign-on model for heterogeneous applications that is based on assistant technology and authentication data centralized managed.

Keywords: single sign-on; heterogeneous application; message mechanism; browser plug-in

1　引言

一般来讲，每个单独的应用系统都会有自己的安全体系和身份认证系统，独立的系统使得进入每一个应用系统都要进行单独的登录，而且登录使用的账号和密码也不尽相同。例如，在医院的信息化管理中，医生需登录 His 系统来查询病人的病史档案，登录 Lis 系统查询病人的化验报告等。这些应用系统一般都是由不同的厂商开发，因此其密码管理和维护的认证系统也不相同。

为了正常的使用各个应用系统，用户往往需要记住或保存多份账号和密码。多份账号密码很大程度上增大账户管理的难度，增加了用户遗失和被盗密码的几率，琐碎重复的系统登录也给用户工作效率造成一定的影响。当网络覆盖范围不断扩大、应用的访问量越来越大，迫切要求系统对用户、安全、网络等进行统一管理。

为了解决这些问题，研究者提出了 CAS 单点登录(SSO)方案，解决了多个 Web 应用的集成问题。但是在针对 C/S 与 B/S 模式并存及不允许修改的异构应用系统上，并没有很好的解决方案，而这样的情况在实际应用中却是普遍存在的。本文将在基于对当前 SSO 解决方案的分析基础上，结合客户端 Windows 系统的窗口 Hook 技术以及浏览器 BHO 技术，对异构系统的单点登录方法进行研究，并提出相应的解决方案。

① 收稿时间:2009-12-31;收到修改稿时间:2010-01-31

© 中国科学院软件研究所 http://www.c-s-a.org.cn

2 单点登录概述

单点登录技术是一种认证和授权机制，它允许用户只用通过一次验证就可以访问所有的授权的被信任的应用系统，即当用户切换使用到另外的系统时，不需要重新登录。在多个应用系统中，单点登录是应用系统整合的重要基础之一。

2.1 CAS 单点登录

CAS (Central Authentication Service)是目前应用最广泛的针对 Web SSO 的解决方案[1]，基于计算机网络授权协议 Kerberos[2]。CAS 由二个部分组成：

(1) CAS 服务器

负责用户的身份验证并传送验证结果返回，以及相应 CAS 客户端的认证信息请求。

(2) CAS 客户端

负责重定向未验证身份的用户到CAS服务器和响应已验证用户的请求。CAS 客户端必须是使用了 CAS 客户端库的Web应用，或者是以CAS模式运行的Web应用服务器。

即当用户切换使用到另外的系统时，不需要重新登录。在多个应用系统中，单点登录是应用系统整合的重要基础之一。

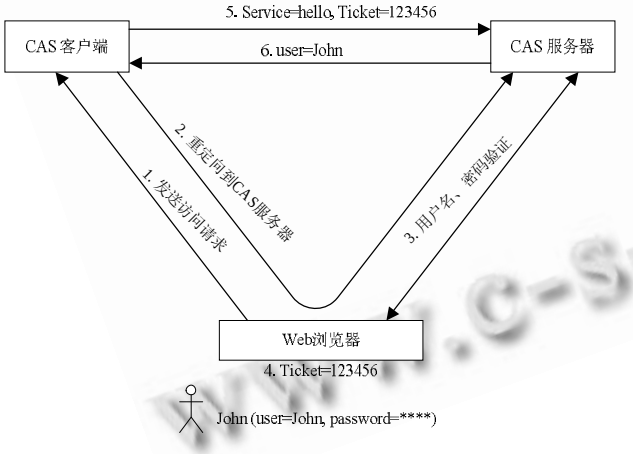

图 1 CAS 单点登录流程示意图

当用户John打开Web浏览器登录CAS客户端上的 hello 应用时，请求被立即重定向到 CAS 服务器进行认证，重定向对于用户是透明的。认证完成后，用户浏览器得到与 CAS 相同的票据，以 Cookie 方式存储以此作为已认证的标识。当 John 需要访问另一个 hello2 应用时，CAS 客户端把从用户浏览器得到的票据和 CAS 服务器得到的票据比较，核实用户身份合法性。核实后，CAS 客户端将相应应用的处理结果发送回用户，完成整个请求过程。

CAS 可以方便的解决多个 Web 应用的单点登录，但同时也暴露出其局限性：

(1) 不能保证原有 Web 应用的独立性，Web 应用若提供单点登录，则需进行验证模块的修改。

(2) 验证票据使用浏览器 Cookie 作为存储媒介，存在安全隐患。

(3) CAS 单点登录只能解决多个 Web 应用的单点登录，不能解决其他系统结构的应用，如 C/S 应用。

3 异构应用的单点登录

由于异构应用客户端与服务端通信的方式各不相同，采用服务器端验证模块整合的方式复杂且难度大[3]。

分布式的应用一般采用 C/S 或 B/S 架构，从客户端的角度分析发现，C/S 架构的客户端是窗口程序，B/S 架构客户端是浏览器。窗口程序和浏览器客户端通过相应的技术进行控制，使得外部辅助程序能够通过控制客户端完成登录信息的代理输入，从而实现无重复的登录。

3.1 C/S 辅助程序与 Windows 消息机制

作为 C/S 架构客户端的窗口程序，底层由系统库函数实现，它的特性与系统密切相关。Windows 是消息驱动式系统，各个应用程序之间以及应用程序与系统之间的通信都是由 Windows 消息来作为媒体。应用程序中的功能由消息触发来驱动，消息的响应和处理实现功能的交互[4]。

Windows 消息系统有消息队列、消息循环及窗口过程三部分组成：

(1) 消息队列是一个先进先出的消息序列。消息队列分为系统消息队列和应用程序消息队列，所有的被触发的消息都首先被放入系统消息队列中，再由系统将消息拷贝到消息所属的应用程序队列中。

(2) 消息循环是应用程序从对应的消息队列中检索出消息、分派该消息到相应窗口的循环过程。

(3) 窗口过程是由 Windows 系统定义的回调函数，它的任务是获取或接受系统传递给窗口的消息并响应和处理该消息，最后给 Windows 一个返回值。

在 Windows 进程通信中，一个进程或线程可以向另一进程或线程的消息队列发送消息，指定另一进程或线程的行为，这样达到进程或线程间的通信。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

进程 A 通过登录窗口进行用户登录的过程如下：

(1) 用户点击登录后，系统创建点击操作的鼠标消息；

(2) 消息进入消息队列，消息队列按序取消息；

(3) 操作系统将消息发送给对应的窗口过程，窗口过程完成对应的逻辑处理，如连接服务器验证用户名密码的操作。

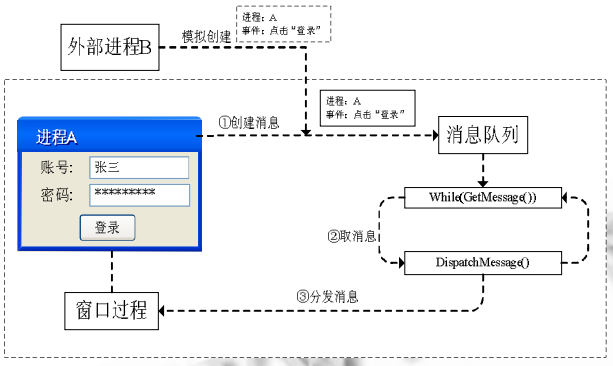

图 2　外部进程指定进程行为

进程 A 登录的过程，也可以通过外部进程 B 来完成。进程 B 创建消息并发送到进程 A 的消息队列中，消息的内容是进程 A 里窗口中“登录”按钮所对应的点击事件。通过消息的循环和分发，消息传递给窗口过程，从而指定进程 A 完成点击“登录”按钮，进行登录过程。

3.2 B/S 辅助程序与浏览器插件扩展 BHO 技术

B/S 架构的分布式系统以浏览器作为客户端，HTML 作为信息传输的载体。用户使用浏览器向服务器发出请求，服务器响应并将响应结果以 HTML 文本形式发送给浏览器。HTML 标准定义了一系列的控件标签，包括按钮、文本框，表格，图片等，通过浏览器的对这些标签以及其它 HTML 内容的解析渲染，最后呈现的图像界面。

浏览器呈现界面控件与操作系统控件有本质区别。HTML 控件由浏览器程序渲染的，而一般应用程序的控件是由操作系统渲染的。另外 HTML 一种通用的标记语言，独立于平台系统，因此通过操作系统的消息方式不能对浏览器内应用界面进行控制。

IE 是目前最广泛使用的浏览器，其通过 BHO（Browser Helper Object）提供了开放交互的浏览器接口。符合 BHO 接口标准的程序代码被以 DLL 动态链接库形式在注册表里注册为 COM 对象，还要在 BHO 接口的注册表入口处进行组件注册，以后每次 IE 启动时都会通过这里描述的注册信息调用加载这个 DLL 文件，而这个 DLL 文件就因此成为 IE 的一个模块（BHO 组件），与 IE 共享一个运行周期，直到 IE 被关闭。浏览器允许 BHO 访问当前页面的文件对象模型（DOM）及控制浏览器的导航，因此恰当的实现 BHO 接口，就可以对当前的登录界面进行控制。

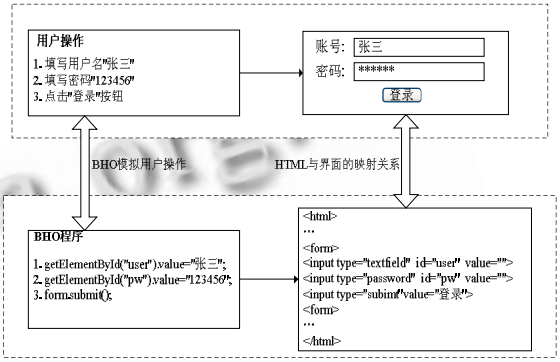

图 3　BHO 模拟用户登录操作

通过对登录页面 HTML 对应的 DOM 对象的控制，修改 DOM 中的对应的用户名和密码的字段值来完成代理输入认证信息的过程，然后调用 DOM 函数，完成认证信息的提交从而实现辅助登录。

3.3 单点登录的系统架构

在异构应用系统的架构中，用户对应多个应用系统。将多个应用系统的认证信息集成并统一到一个本地设定的认证身份上，使用此设定的认证身份进行所用应用系统的登录。

系统主要由认证信息池、Manager 和辅助程序三个部分组成。

所有应用的认证信息都被保存在认证信息池 LDAP[5]目录服务器中。LDAP 目录服务器可以方便地进行用户的角色权限的设置。认证信息池中的角色分为一般用户与超级管理员，超级管理可以对所有用户信息进行修改配置，一般用户只能修改个人信息。LDAP 目录是树状结构，应用系统认证信息组织在一个树的节点上，这样就可以方便地进行用户的角色权限的统一设置。

辅助程序解决了重复登录的问题。在上两节中已经分别对 C/S 客户端和 B/S 客户端进行剖析，并提出了针对各自特点使用外部程序进行代理登陆的方案。只用提供两个不同的辅助程序，不需要每一个应用都

© 中国科学院软件研究所 http://www.c-s-a.org.cn

使用单独的辅助程序。辅助程序与 Manager 进行交互，读取 Session 中的不同应用系统的认证信息，并协助完成自动登录。辅助程序部署在操作系统或浏览器的启动加载库中，在应用启动或页面加载时触发。

Manager 是整个系统的中心控制器，保存和维护 Session，负责与辅助程序的交互，提供认证信息池的内容进行维护和管理的接口。用户登录后，Session 中 Current 设置用户唯一标识，与 Current 关联的认证信息池中的数据将被读取供辅助程序使用。

Manager 提供用户登录的公共界面，用户登录 Manager 后则不需要在进行其他应用程序界面的登录，这样就实现了单点登录。

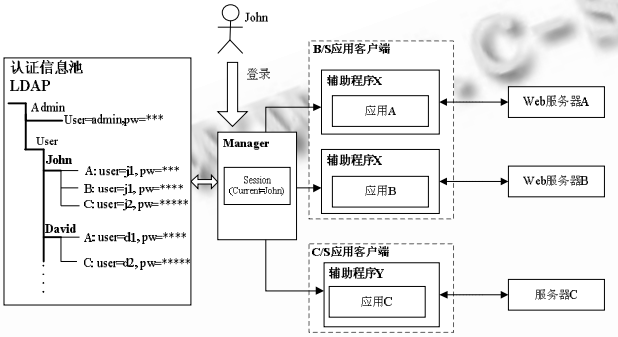

图 4 系统结构

用户 John 使用应用 A 和 C 访问外部资源分为以下 9 个步骤:

(1) 用户打开 Manager 界面，输入用户名 John 和密码。

(2) Manager 查询 LDAP 验证登录信息是否合法，通过验证后，调取 John 目录的子树的内容并缓存。

(3) Manager 隐藏界面，并以系统服务方式运行。

(4) 用户运行应用 A。

(5) 辅助程序 X 新实例被创建启动，之后从 Manager 中读取 A 对应的认证信息(user=j1，pw=***)，并代理应用 A 录入用户名密码。

(6) A 的认证信息通过了 Web 服务器的验证，完成 A 应用的登录。

(7) 用户运行应用 C。

(8) 辅助程序 Y 新实例被创建启动，并从 Manager 中读取 C 对应的认证信息(user=j2，pw=*****)，代理应用 C 录入用户名密码完成登录。

(9) C 的认证信息通过了服务器的验证，完成 C 应用的登录。

登录 Manager 之后，用户 John 的操作仅仅是运行 A 和 C 的应用即可登录。用户并的登录由辅助程序 X、Y 协助，其中的取密码、以及登录信息的录入，这对于用户都是透明的。另外如果登录失败，可能由用户认证信息错误引起，Manager 提供维护密码的接口，可以对密码进行重新配置。

4 安全与性能分析

安全性是单点登录的关键要素。异构应用的解决方案中，密码保存在认证信息库 LDAP 中，能保证各应用系统的认证信息的安全不受威胁。LDAP 可以选择多种加密方式对认证信息进行加密，用户名密码以加密字符串的方式存储在 LDAP 库中，外部程序与 LDAP 的连接通过 LDAP 提供的 SASL 安全验证。另外，用户名与密码的维护通过 Manager 提供的接口进行修改。辅助程序的代理登陆也消去了因操作失误带来的安全隐患。

5 结束语

本文针对异构的系统应用提出了基于客户端的单点登录解决方案，在客户端技术上引进了消息机制以及浏览器插件技术实现辅助程序的代理登录，在系统架构上通过 LDAP 保证认证信息的安全性。

今后的研究方向集中于对多种客户端的技术分析，提高辅助程序的稳定性。通过对辅助程序的扩展，丰富代理操作的类型，实现多应用之间的协同工作。

参考文献

1 ESUP-Portail: Open source Single Sign-On with CAS. http://perso.univ-rennes1.fr/pascal.aubry/present-ations/ca seunis2004.

2 周倜,王巾盈,李梦,等. Kerberos 协议版本的分析与比较.计算机科学, 2009,36(2):119－121.

3 王翔宇,刘高嵩,龙军.异构系统的信息协同机制的研究.计算机系统应用, 2009,18(5):18－21.

4 王强,周明,李定国. Windows API for 2000/XP 实例精解.北京:电子工业出版社, 2002.

5 肖琬蓉,杨生举.基于 LDAP 的统一用户认证系统设计与实现.计算机科学, 2008,35(5):298－300.

© 中国科学院软件研究所 http://www.c-s-a.org.cn