第 34 卷 第 8 期 计 算 机 工 程 2008 年 4 月
Vol.34 No.8 Computer Engineering April 2008
·安全技术· 文章编号：1000—3428(2008)08—0179—03 文献标识码：A 中图分类号：TP393.08

单点登录在 Web 服务安全中的应用

王 茜，吴黎明
(重庆大学计算机学院，重庆 400044)

摘 要：针对目前单点登录应用于 Web 服务安全时存在的问题，结合 WS-Security 和 SAML 规范提出一个 Web 服务身份认证和授权的单点登录模型，描述该模型的单点登录过程及实现，对其安全性进行了分析并给出了相应的安全策略。该系统模型具有兼容性、容易部署及良好的可扩展性等特点。

关键词：单点登录；Web 服务；安全断言标记语言；Web 服务安全性

Application of Single Sign-on in Web Services Security

WANG Qian, WU Li-ming
(School of Computer Science, Chongqing University, Chongqing 400044)

【Abstract】 Aiming at current existed problems of single sign-on applied to Web services security, this paper puts forward a single sign-on model for Web services authentication and authorization with the help of WS-Security and SAML, and analyzes the flow and security of this single sign-on system. The system has compatibility and better expansibility it can be deployed easily.

【Key words】 single sign-on; Web services; Security Assertion Markup Language(SAML); WS-security

1 概述

Web 服务是由 URI 标识的软件应用程序，其接口和绑定可以通过 XML 构件进行定义、描述和发现。它提供了一种使用 HTTP、SMTP 或 FTP 等 Internet 兼容协议来访问业务或者应用程序逻辑的方式。Web 服务是一种有效的解决方案，具有很好的互操作性和可扩展性，克服了以往 Web 应用缺少交互和通信的弱点，可实现客户和企业之间快速而灵活的信息共享。Gartner 预测 Web 服务将在近几年成为企业部署新应用程序解决方案的首选技术。所以，Web 服务的安全性问题也越来越受到人们的关注。

Web 服务的安全包括身份认证和访问控制、传输双方的加密与解密以及处理来自于不同目的和不同组织的网上威胁等。其中，身份认证和访问控制是保障以安全、有效的方式访问 Web 服务受保护的前提，在 Web 服务系统安全设计中占有重要地位。然而 Web 服务分布式、异构的本质，使得对服务请求者进行身份认证和授权变得更加复杂。单点登录(Single Sign-On, SSO)提供了一种很好地解决该复杂问题的方法。

单点登录，也称单一登录，通常指一个用户在使用多个应用时只需要同一个认证信息(比如：用户名/密码)，并且只需要登录一次就可使用所有的应用。许多公司(包括 IBM、微软、惠普、Sun 等)和组织都对单点登录作了研究，并取得了很多成果。

微软提出了.NET Passport解决方案[1]。Passport是一个单点登录服务，它采用一种集中的方法来处理跨域SSO。如果一个用户希望访问一个支持Passport的站点，这个用户会被重定向到一个.NET Passport登录服务器。一旦该用户成功地通过了身份认证，加密的身份认证数据就被作为查询字符串参数追加到URL，而且这个用户会被重定向到最初访问的站点，一个被称为.NET Passport管理器的COM组件会解密这个身份认证信息，在站点内验证这个用户的身份，并使用它来授权访问受保护的资源。

自由联盟计划(liberty alliance project)[2]是由 Sun Microsystems 与其他公司建立的一个联合会，它制定了定义联盟标识的一种规范，提倡一种开放的分散系统，而且能够容纳多种身份认证机制。Liberty的目的是允许不同的Web站点代表其客户进行合作，从而促进更多无缝和直观的活动。联盟是Liberty Alliance采用的一种方法，它表示可以在一组信任的对等实体之间共享身份认证信息，从而能够实现SSO。Liberty建立在OASIS SAML规范的基础上，并扩充了OASIS SAML规范。Liberty利用SAML身份认证断言实现了在信任圈成员内的单点登录。

这些单点登录解决方案能够很容易在Web站点身份认证中被应用，但是它们都不能完全适用于 Web 服务的身份认证和授权。比如：Passport 的封闭性和隐私性因素一起成为它被更加广泛接受的障碍，并且目前的 Passport 不支持 Web 服务；自由联盟计划要求实体之间互相信任，这在现实的 Web 服务环境中很难满足。

2 Web 服务应用中的身份认证和授权要求

Web 服务提供商(eService*X*.cn, *X*=1,2,…)提供各式各样的个性化 Web 服务，不同提供商提供不同的服务；用户可能需要调用多个提供商提供的 Web 服务。

为了处理和实现一个 Web 服务请求，需要满足以下几个条件：

(1)eService*X* 必须可靠地知道请求的发起者是谁；

(2)eService*X* 希望保证请求发起者具有访问的授权；

(3)eService*X* 希望尽可能多的用户使用自己的 Web 服务；

作者简介：王 茜(1964－)，女，副教授、博士，主研方向：网络安全，电子商务和远程教育；吴黎明，硕士研究生

收稿日期：2007-04-30 **E-mail：**wangqian@cqu.edu.cn

(4)eService*X* 不希望自己维护一个庞大而多变的用户数据库；

(5)用户不希望在每一个 Web 服务提供商网站提交个人信息。

一个受信任的第三方，即单点登录服务，可以帮助解决这些问题。受信任方(源站点，WebSSO.cn)用自己的数据库验证用户的身份，然后把身份认证的证据传递给 eService*X*(目标站点)。在这种情况下，eService*X* 不需要重复认证用户的身份，而是要验证用户信息来自于一个值得信赖的来源(即WebSSO.cn)，它可以保证身份认证信息的真实性。

3 应用于 Web 服务安全的单点登录

3.1 WS-Security 和 SAML

WS-Security[3]定义了一个抽象化安全服务的单一安全模型。在该模型中，请求者向Web服务申请资源。Web服务使用断言来作出与安全相关的决定，即在满足这个申请之前，Web服务要求提供断言的证据。这些断言可以是一个身份或者一个许可。如果请求者具备需要的证据，这个证据将会在一个安全令牌中发送给Web服务。如果请求者不具备这个证据，服务的提供商将尝试从一个安全令牌服务获取这个证据。WS-Security规范中广泛利用XML签名和XML加密来保证Web服务消息的完整性和机密性。

WS-Security 当前的最新版本是 Web Services Security v1.1，由 OASIS 于 2006 年 2 月发布。

安全断言标记语言(SAML)[4]是一种规范，它定义了表示身份认证、属性和授权信息的一种标准方法，在分布式环境中，各种不同的应用程序都可以使用这些身份认证、属性和授权信息。所有遵守SAML规范的安全服务，都可以解释从一个安全服务发送到另一个安全服务的安全数据。

SAML 规范的核心是定义安全数据表示形式的 XML 模式，即断言。除了定义断言以外，SAML 还描述了应用程序如何通过绑定和配置文件来传输断言。绑定描述了第三方安全服务请求和传递断言的方式，以及如何在一个消息协议中传递断言。配置文件定义了使用 SAML 断言来支持应用程序之间事务安全的方法。

SAML 当前的最新版本是 SAML v2.0，由 OASIS 于 2005 年 3 月发布。

3.2 单点登录过程

WS-Security 定义 SOAP 消息的一个安全元素，安全元素中包含所有的声明。SAML 在解决安全的 Web 服务互操作性的过程中能够起到重要作用。访问 Web 服务的主体可以是浏览器，也可以是一个特定的应用程序。本文选用浏览器访问 Web 服务的方式，采用 SAML 断言作为安全信息定义的标准化格式，通过 SOAP 消息传递安全元素。

用户在 SSO 注册中心注册，获取由 SSO 签发的个人身份标识文件(用户名/密码)。用户访问所有信任 SSO 源站点的目标站点时的单点登录过程如图 1 所示。

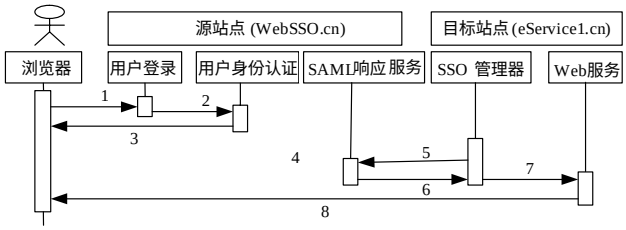

图 1 应用于 Web 服务安全中的单点登录过程

(1)用户浏览器访问源站点(WebSSO.cn)，该过程可以是用户直接输入源站点网址，也可以是用户请求目标站点(eService1.cn)受保护的 Web 服务被目标站点重定向到源站点。此时，源站点要求用户登录以确认用户身份。

(2)用户提交身份信息，源站点上用户身份认证服务确认用户的合法性。

(3)确认用户身份成功后，源站点给用户浏览器返回一个 SAML Artifact 文件，表现形式为目标站点服务资源的 URL 发生改变，即在 URL 参数中增加 Artifact 名值对。

(4)用户点击相应超链接以选择想要的服务资源，用户浏览器重定向到目标站点。

(5)目标站点的单点登录管理器收到步骤(4)传递来的信息后，从 Artifact 中解析出源站点信息，然后单点登录管理器向源站点(WebSSO.cn)的 SAML 响应服务发送 SAML 请求，该请求中包含了 Artifact 信息。

(6)源站点的 SAML 响应服务接收到目标站点传递来的 SAML 请求后，从请求信息中包含的断言引用信息找到相关断言，然后将断言封装在 SOAP 包中以 SAML 响应方式传送到目标站点单点登录管理器。

(7)目标站点单点登录管理器检查用户断言信息，检查成功通过后则将用户重定向至 Web 服务资源所在 URL。

(8)目标站点将用户所需资源发送到用户浏览器。

若用户还需访问信任域中其他目标站点(如 eService2.cn)上的 Web 服务资源，此时不需要再次登录，访问过程只需步骤(4)~步骤(8)。

3.3 单点登录系统实现

上述单点登录系统主要由 4 个部分组成：用户注册，身份认证服务，SAML 响应服务和单点登录管理器。其中，前 3 部分位于源站点，单点登录管理器部分则是常驻目标站点的 HTTP 模块。HTTP 模块可以对一个请求进行预处理和后处理，它截获并处理系统事件以及其他模块产生的事件。

系统中各个部分的核心程序，如表 1 所示。

表 1 单点登录系统核心程序

所属部分	核心程序名称	实现功能	备注
用户注册	Register.aspx	新用户注册	ASP.NET 页面
身份认证服务	Logon()	验证用户名/密码	Web 服务 UserLogin.asmx
	MakeSAML()	产生 SAML 断言，并返回 Artifact	
	AppendArtifact()	在目标 Web 服务资源 URL 上附加 Artifact	
SAML 响应服务	SAMLReciever()	接收目标站点的 SAML 请求，从中提取 Artifact 信息	Web 服务 AAS.asmx
	AssertionRetrival()	通过 Artifact 检索对应 SAML 断言	
	SAMLResponder()	发送 SAML 断言	
单点登录管理器	ArtifactConsumer()	获取用户的 Artifact	HTTP 模块
	SAMLRequester()	使用获取的 Artifact 索引向源站点发送身份验证请求，并接收响应	
	AssertionConsumer()	验证 SAML 断言并实施授权	

3.4 安全性分析

单点登录过程需要在不同域间、应用程序之间不断地交换安全信息，因此，单点登录系统的安全性至关重要。下面针对 Web 应用安全中常见的攻击手段给出相应的安全策略：

(1)抗拒绝服务攻击。采用集中式身份认证，单点登录服务器比较容易遭受拒绝服务攻击。而本文中用户身份认证、SAML 响应服务均以 Web 服务的方式提供，服务的 URL 可以不暴露给用户。另外，把可以发送 SAML 请求消息的用户限制在特定范围(信任域内的合作站点)，也可以在一定程度上减少拒绝服务攻击。

(2)抗重放攻击。在 SAML 请求/应答消息中使用时间戳和唯一的 Artifact 标识，可以防止重放攻击。

(3)防止消息的篡改。使用 WS-Security 中的 XML 签名机制对 SOAP 消息进行数字签名，以保证其完整性。使用 WS-Security 中的 XML 加密机制对 SOAP 消息中的敏感数据进行加密，以保证其机密性。

(4)抗中间人攻击。中间人攻击是指攻击者截获通信双方的报文并用自己的报文替代原始报文。使用 SSL 对传输数据进行签名和加密，可以防止中间人读取会话内容，从而阻止传输数据的中间人攻击，同时保证会话密钥的安全性。

4 相关工作

国内对 Web 服务的单点登录方面开展了一些研究，取得了相应的研究成果，并提出了一些解决方案。文献[5]提出了一个基于 XKMS 和 SAML 的 Web 服务安全模型，它使用 XML 密钥管理规范(XKMS)保存用户的注册信息并负责验证用户登录，然后利用 SAML 服务实现了单点登录的功能，该模型注重对用户凭证的保护，需要在对用户实施身份认证的 Web 服务站点中增加额外的特殊服务。文献[6-7]提出的实现 Web 服务的单点登录方法都是基于 SAML 的，在这些单点登录系统的体系结构中，Web 服务站点既充当服务提供者又充当身份提供者角色(用户访问其他 Web 服务站点时)，这将需要对 Web 服务站点做出修改。文献[8]提出了另一种基于 SAML 的单点登录系统，其安全令牌存储在客户端，包含在 SOAP 头中传递，被调用的 Web 服务从 SOAP 中获得 SAML，并且使用 WASP Card 证书对其进行验证，因而该系统只能用于有相应约定的 Web 服务环境。

本文提出的应用于 Web 服务的单点登录系统，以一个受信任的第三方服务出现，支持多种身份认证方式。单点登录管理器使用 HTTP 模块技术开发，可以在无需对 Web 服务修改的情况下部署并对 Web 服务实施安全保护，并且通过修改单点登录管理器的配置，可以使其与多个具有不同 URL 地址的单点登录服务通信。因此，该系统具有兼容多种身份认证方式、容易部署、能够免除服务提供方安全基础设施的重复建设以及具有良好的可扩展性等优点。

5 结束语

单点登录技术使用户在使用多个 Web 服务时无需进行多次登录，从而可以更加方便、安全地管理用户账户。WS-Security 定义了把安全信息附加到 SOAP 消息的标准方法，而 SAML 则定义了用来交换身份认证、属性和授权断言的一种格式。结合这两种规范得出的单点登录系统，提供了对安全上下文的描述，为异构安全系统间的安全信息互操作带来可能，使得用户可以方便地访问异构安全系统中的 Web 服务资源。

参考文献

[1] Passport Single Sign In. MSDN[DB/OL]. (2006-12-20). http://msdn.microsoft.com/library/en-us/passport25/.

[2] Liberty Alliance Project. LAP[EB/OL]. (2006-12-20). http://www.projectliberty.org/.

[3] WS-Security v1.1. OASIS[DB/OL]. (2006-12-20). http://www.oasis-open.org/committees/documents.php?wg_abbrev=wss.

[4] OASIS Security Services TC. OASIS[EB/OL]. (2006-12-20). http://www.oasis-open.org/ committees/tc_home.php? wg_abbrev=security.

[5] 韩 涛, 郭荷清. Web 服务安全模型的研究与实现[J]. 计算机工程, 2006, 32(10): 130-134.

[6] 钟迅科. 基于 SAML 实现 Web 服务的单点登录[J]. 现代计算机, 2004, 21(4): 32-36.

[7] 余 荣, 刘明华. 基于 SAML 实现 Web Service 的单点登录[J]. 计算机与现代化, 2005, 21(12): 81-85.

[8] 韩 伟, 范植华. 基于 SAML 的单点登录技术在 Web 服务中的应用研究[J]. 计算机工程与设计, 2005, 26(3): 634-636.

(上接第 178 页)

tag，因此协议仍然可能受到类型漏洞攻击。(2)实际中使用的很多协议，如 GDOI, Kerberos 是比较复杂的，对于这些复杂协议，添加 tag 会给协议处理带来很大的额外开销。因此，tag 方法对于一些嵌入式系统，如 PDA、传感器网络、智能卡系统等，明显是不合适的。(3)如果被解密数据存在可被判断的标记，那么就增加了被破译的危险，这一点在使用 tag 方法时是需要慎重考虑的。

2.3 检查消息长度防止类型漏洞攻击

类型漏洞攻击的特点是利用不同数据在表示方式上的相似性，以一种数据或几种数据字段充当其他数据来欺骗通信主体。笔者认为可以在协议约定中严格定义各种数据的表示方式和长度，并避免不同数据具有相同位长度或者不同数据长度之和等于另一数据的长度。定义之后，在协议实现时，预先计算各消息长度或消息中加密部分的长度，然后在处理接收到的消息时，将消息实际长度与预先计算值相比较，如果不相符，说明存在类型漏洞攻击，则丢弃该消息。

该方法从类型漏洞攻击本质出发，可以防止类型漏洞攻击的发生，但同时也会增加协议消息处理的开销。

3 结束语

本文对类型漏洞攻击进行了探讨，根据类型漏洞攻击的特点，提出可以通过检查消息长度防止类型漏洞攻击的方法，并结合 GDOI 遭受类型漏洞攻击的实例，指出了 tag 方法的局限性。

在安全协议的设计中，首先应了解类型漏洞攻击的特点和危害，避免协议中加密消息类型相同或相似。其次可通过形式化的方式来检测和验证协议中是否存在类型漏洞攻击以进行改进。这样才能在不增加协议处理额外开销的条件下，最大程度上避免类型漏洞攻击的发生。

参考文献

[1] 侯俊峰. 安全协议形式化验证和安全协议设计研究[D]. 北京, 中国: 清华大学, 2004.

[2] Woo T Y C, Lam S S. A Lesson on Authenticated Protocol Design[J]. Operating Systems Review, 1994, 28(3): 24-37.

[3] Baugher M, Hardjono T, Harney H, et al. The Group Domain of Interpretation[EB/OL]. (2001-05-30). http://tools.ietf.org/html/draft-ietf-msec-gdoi-01.

[4] Meadows C, Syverson P, Cervesato I. Specification and Analysis of the GDOI Protocol Using NPATRL and the NRL Protocol Analyzer[J]. Journal of Computer Security, 2004, 12(6): 893-931.

[5] Heather J, Lowe G, Schneider S. How to Prevent Type Flaw Attacks on Security Protocols[C]//Proc. of the 13th IEEE Computer Security FoundationsWorkshop. [S. 1.]: IEEE Computer Society Press, 2000: 255-268.