基于属性的 Web 服务访问控制模型

傅鹤岗　李　竞
（重庆大学计算机学院　重庆 400030）

摘　要　传统访问控制模型都是静态的、粗粒度的，不能很好地在面向服务的环境中应用。本文提出了一种基于属性的访问控制模型(ABAC)，它结合 SAML(Security Assertion Markup Language，安全声明标记语言)和 XACML(Extensible Access Control Markup Language，可扩展访问控制标记语言)标准，能够基于主体、客体和环境的属性来动态地、细粒度地进行授权。新的模型更加灵活，特别适用于动态的 Web 服务环境。

关键词　Web 服务，访问控制，SAML，XACML

Attribute-based Access Control Model for Web Services

FU He-Gang　LI Jing
(Computer College, Chongqing University, Chongqing 400030)

Abstract　Traditional access control models are not well-suited for the services-oriented environments because they are mostly static and coarsely grained. An Attribute-based access control (ABAC) model is proposed based on SAML and XACML, which adopted the authorization mechanism dynamically and fine-grained based on subject, resource and environment attributes. The new model is more flexible, which is especially suitable for the dynamic, ad-hoc environment for Web services.

Keywords　Web services, Access control, SAML, XACML

1　引言

Web 服务作为一种新的分布式计算模型，能够在各种异构平台的基础上构筑一个通用的、与平台无关的、与语言无关的技术层，使各种不同平台之上的应用方便地实施连接和集成。Web 服务具有简单、跨平台、松散耦合的特点，它将成为下一代电子商务的框架。由于完全采用开放的标准协议，Web 服务很容易受到安全性方面的威胁。然而 Web 服务与传统电子商务环境相比具有更强的分散性和动态性，传统的安全机制并不能有效地解决 Web 服务所面临的问题。目前安全问题严重制约了 Web 服务的广泛普及，因此，研究 Web 服务环境下访问控制模型具有重要的意义。

在面向服务的环境中，信息的访问通常是动态的。传统电子商务环境下的访问控制模型，如访问控制列表(ACL)、基于角色的访问控制(RBAC)等，都是静态的、粗粒度的，所以它们并不能很好地在面向服务的环境中应用。本文针对 Web 服务环境的特点，提出了一种基于属性的访问控制模型(ABAC)，它结合 SAML(Security Assertion Markup Language，安全声明标记语言)和 XACML(Extensible Access Control Markup Language，可扩展访问控制标记语言)技术，基于主体、客体和环境的属性来动态地、细粒度地进行授权。

本文接下来将先分析现有访问控制模型在 Web 服务环境下的局限性，然后详细介绍新型的基于属性的访问控制模型(ABAC)以及 SAML、XACML 技术在 ABAC 模型中的应用，最后介绍如何应用开源工具来实现 Web 服务的 ABAC 模型。

2　现有的访问控制模型

访问控制是指通过某种途径显式地准许或限制访问能力及范围的一种方法。通过访问控制服务，可以限制对关键资源的访问，防止非法用户的侵入或者因合法用户的不慎操作所造成的破坏。传统的访问控制类型主要有两类：自主访问控制 DAC (Discretionary Access Control)和强制访问控制 MAC (Mandatory Access Control)。

随着企业信息系统的迅速发展，对访问控制服务的质量也提出了更高的要求，以上两种访问控制技术已很难满足这些要求。DAC 将赋予或取消访问权限的一部分权力留给用户个人，这使得管理员难以确定哪些用户对哪些资源有访问权限，不利于实现统一的全局访问控制。而 MAC 由于过于偏重保密性，对其他方面如系统连续工作能力、授权的可管理性等考虑不足。20 世纪 90 年代以来出现的一种基于角色的访问控制 RBAC(Role-Based Access Control)技术有效地克服了传统访问控制技术中存在的不足之处，可以减少授权管理的复杂性，降低管理开销，而且还能为管理员提供一个比较好的实现安全政策的环境[1]。RBAC 的核心思想就是将访问权限与角色相联系，通过给用户分配适合的角色，让用户与访问权限相联系。角色是根据企业内为完成各种不同的任务需要而设置的，根据用户在企业中的职权和责任来设定他们的角色，用户可以在角色间进行转换，系统可以添加、删除角色，还可以对角色的权限进行添加、删除。RBAC 将安全性放在一个接近组织结构的自然层面上进行管理。

以上这些访问控制模型虽然在某些情况下非常有效，但并不能完全满足 Web 服务环境的要求，主要表现在以下两个方面：

首先，传统信息系统只需要处理系统已知的用户，用户也只访问已知的系统。然而在 Web 服务环境下，服务消费者与服务提供者通常来自不同的域，彼此是陌生的，他们之间关系

傅鹤岗　副教授，硕士生导师，主要研究方向：软件工程、电子商务。李　竞　研究生，主要研究方向：电子商务、Web 服务。

的建立是临时的、动态的。服务消费者可以通过网络发现新的服务提供者，实时地访问其数据。所以对 Web 服务而言，无法提前判断什么人会在什么时候访问什么数据。

其次，目前的大多数安全模型都是静态的、粗粒度的，无法为 Web 服务访问控制策略提供丰富的语义支持。以 RBAC 为例，当业务场景越发复杂的时候，就只有建立更多的角色，维护更多的用户-权限关系，大大加重了系统管理的负担。

3 基于属性的访问控制模型(ABAC)

与基于角色的访问控制模型(RBAC)不同，本文提出的基于属性的访问控制模型(ABAC)是根据参与决策的相关实体的属性来进行授权决策的。这里提到的实体分三类：主体(Subject)、资源(Resource)和环境(Environment)。主体属性通常包括身份、角色、年龄、职位、IP 地址等；资源属性主要包括资源的标识、位置(URL)、创建者、创建日期等；环境属性通常描述事务处理时的上下文，包括时间、日期、系统状态、安全级别等。

如上所述，ABAC 是否允许一个主体 s 访问资源 r 是根据主体 s、资源 r 以及当前上下文环境 e 的相关属性来决定的。我们分别用 Attr(s)、Attr(r)和 Attr(e)来表示主体 s、资源 r 和环境 e 的所有属性的赋值。我们也可以通过等号形式来表示单个属性的赋值，如 Role(s)=admin 表示主题 s 的角色是 admin。

通常，ABAC 模型的策略表现为以下形式：

$$can\_access(s,r,e) \leftarrow f(Attr(s),Attr(r),Attr(e))$$

其中，access 通常代表为一系列操作，如 read、write 等；f 是一个布尔函数，它通过主体 s、资源 r 和环境 e 的属性值来计算，其结果决定在当前环境 e 下，主体 s 是否能够访问资源 r。f 函数是根据业务规则来制定的，它描述了主体访问资源时必须满足的条件。

例如，允许系统管理员 admin 对文件 users.dat 进行写操作，就可以建立策略如下：

$$can\_write(s,r,e) \leftarrow (Role(u)=admin) \wedge (URL(r)=http://cqu.cn/uers.dat)$$

从这个例子可以看出，RBAC 模型同样可以用 ABAC 模型很轻松地实现，但后者能够比前者表现语义更丰富的访问控制策略。当业务逻辑发生变化时，RBAC 模型往往需要定义新的角色和权限，而 ABAC 模型仅需按照现实中的语义对策略规则进行修改，所以 ABAC 模型比 RBAC 模型拥有更高的灵活性和更细的访问控制粒度。

4 Web 服务的 ABAC 体系架构

由于 Web 服务是基于 XML 的，我们要在 Web 服务环境下采用 ABAC 模型就需要引入两个基于 XML 的安全性标准：SAML 和 XACML。在讨论 Web 服务的 ABAC 体系架构前，先介绍以上两个基于 XML 的安全性标准。

4.1 SAML

SAML[4]是由 OASIS(高级结构化信息标准组织)批准，基于 XML 实现 Web 服务安全产品之间互操作的安全访问控制框架体系和协议。它是支持 XML 电子商务的第一个工业标准，它将 S2ML 和 AuthXML 结合起来，为企业之间进行 B2B 和 B2C 业务交易提供共享安全服务的公用语言。

声明是 SAML 的基本数据对象，有三种类型：

(1)认证声明(Authentication Assertion)：描述与身份鉴别事件相关的信息，如认证的机构、方式和有效期等；

(2)属性声明(Attribute Assertion)：包含有关主体的特定信息，如主体的信用限制、访问级别、信用等级或其他合法声明；

(3)授权决策声明(Authorization Decision Assertion)：指明主体可以执行或被授权执行的操作。

声明是由 SAML 授权机构发行的，它被绑定到 SOAP 等开放性协议中，并通过 HTTP、SMPT 等通用通信协议在网上传递，达到信任和授权信息共享的目的，解决跨信任域的身份认证和访问控制问题。

SAML 允许企业及其供应商、客户与合作伙伴进行安全的授权、认证和基本信息交换，而与它们各自采用的软件及硬件平台无关。因此，SAML 将会促进离散的安全系统之间的互操作性，可以跨越企业之间的界线，建立一个安全的协同工作的体系框架。在 SAML 框架下，无论用户使用哪种信任机制，只要满足 SAML 的接口、信息交互定义和流程规范，相互之间都可以无缝结合。

和 SOAP 相同，SAML 是完全基于 XML 语言的，同时 Web 服务安全性规范(WS-Security)[6,8]中也规定了 SAML Token 如何与 SOAP 消息进行绑定，所以 SAML 属性声明作为主体属性的载体是一个很好的选择。

4.2 XACML

XACML[5]定义了一种通用的用于保护资源的策略语言和一种访问决策语言。XACML 提供的策略语言允许管理员定义访问控制需求，以便获取所需的应用资源。语言和模式支持包括数据类型、函数和允许定义复杂(或简单)规则的组合逻辑。XACML 提供的访问决策语言用于描述对资源运行时的请求。当确定了保护资源的策略之后，函数会将请求中的属性与包含在策略规则中的属性进行比较，最终生成一个许可或拒绝决策。

另外，XACML 还定义了问询策略系统的请求/响应格式，可使用户构成一个问询来判断一个动作是否被允许执行，并对结果进行解释。请求和响应格式定义了 PDP (Policy Decision Point，策略决策点)与 PEP (Policy Enforcement Point，策略执行点)之间的接口，PEP 发布请求和处理响应，PDP 根据资源的访问策略、请求者的属性和环境属性等，评估请求并返回一个响应。

与以往的策略语言相比，XACML 具有许多独特的优点：

(1)XACML 是基于 XML 标准的，具有能够同时被人和计算机识别的特点，并且提供了一个标准化的访问控制决策模型，为不同的安全域提供了互操作性；

(2)由于它充分考虑了主体、资源和环境的属性，其资源访问策略是基于主体、资源和环境的属性，而不仅仅是基于请求者的身份，因此可以提供比基于身份的、简单地拒绝或授权访问更细粒度的控制访问机制；

(3)XACML 不但处理授权请求，而且还定义了一种机制，用来创建进行授权决策所需的规则、策略和策略集的完整基础设施。特别地，它可利用规则组合算法将来自不同授权实体的多个策略结合成唯一可用的策略集，为位于分布式系统中的资源进行访问控制决策；而且单个的访问策略可被用

于分布在不同管理域的异构资源。

所以，ABAC 的策略模型完全可以通过 XACML 策略来表现。

4.3 Web 服务的 ABAC 体系架构

我们在 Web 服务的 ABAC 体系架构中采用了 SAML 和 XACML 标准：SAML Token 在 Web 服务请求中作为主体属性的载体；XACML 作为访问控制策略语言。具体的体系结构如图 1 所示。

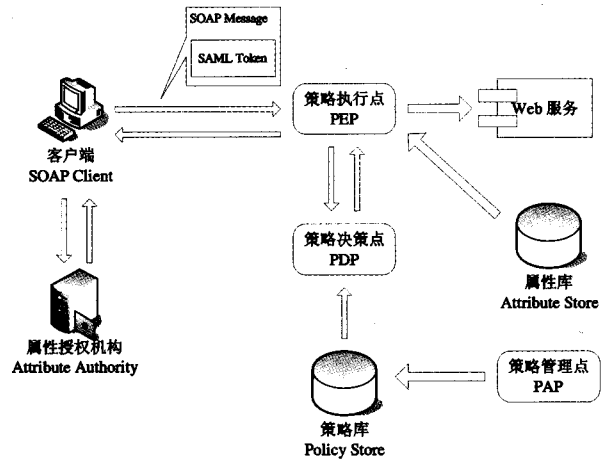

图 1 Web 服务的 ABAC 体系架构

其中，属性授权机构(Attribute Authority)是可信任的第三方认证机构，它主要对主体的相关属性进行确认，声明。主体在通过 SOAP 客户端访问 Web 服务之前，须从属性授权机构获得 SAML 声明，并将获取的 SAML 声明嵌入到 SOAP 消息的头部发送给 Web 服务的访问端。

策略执行点(PEP)处理收到的 SOAP 消息，从消息头部的 SAML 声明中提取主体的属性信息，利用 XACML 请求语言建立一个基于主体、资源、动作和环境等属性的授权请求，并将授权请求发送给策略决策点(PDP)。

策略决策点(PDP)根据接收的 XACML 授权请求从策略库中提取对应的 XACML 策略来判断访问请求是否合法，最后将决策结果(允许或拒绝)返回，并由 PEP 进行访问控制的实施。

策略管理点(PAP)维护策略库，根据业务逻辑定义策略决策点(PDP)所需要的 XACML 策略。

5 实现

本节将介绍如何利用开源工具实现 Web 服务环境下的 ABAC 模型。我们采用 Apache Axis[7] 作为 SOAP 引擎，结合 WSS4J 和 sunxacml 等开源工具完成了 Web 服务环境下 ABAC 模型的一个实现。通过分析 Web 服务的 ABAC 体系架构可以发现，其实现有两个关键：

(1)在客户端，SAML 声明需要按照 WS-Security 规范嵌入到 SOAP 请求消息的头部；

(2)在服务器端，PEP 需要从 SOAP 请求中获取属性，并向 PDP 发出请求，根据响应结果进行访问控制的实施。

5.1 客户端

实现采用了 Apache Axis 作为 SOAP 引擎。Axis 是 Apache 软件基金会开发的一个 SOAP 标准的 Java 实现，是早期 Apache SOAP 项目的继续。由于 Axis 支持很多规范，源代码公开，并且是免费的，因此有相当广泛的用户基础。

Axis 具有的一个关键特点是它为用户提供了模块的即插即用性。几乎 Axis 的任何模块都可以被替换，也可在消息处理流程中任意加入模块，实现此机制最重要的一个概念是 Handler。Handler 是消息流过的一个模块，使用 Handler 可对消息进行自由的处理。

Axis 对消息的处理是通过调用一系列 Handler 来完成的。图 2 是 Axis 在客户端对消息处理的示意图，其中小圆柱表示 Handler。

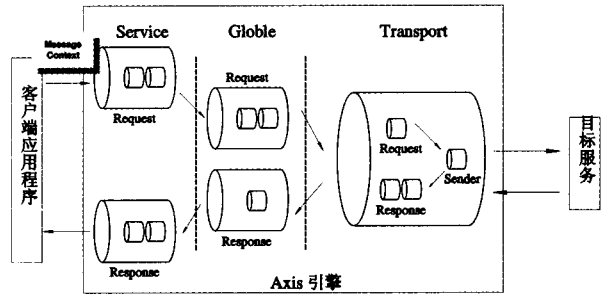

图 2 Axis 客户端消息处理图

Axis 中对 SOAP 消息的处理是通过在 Handler 间传送一个 MessageContext 对象来完成的。MessageContext 含有处理请求消息或响应消息所需的所有数据，其中关键数据有：一个请求消息、一个响应消息以及当前消息处理的所有元数据。每个 Handler 都可以访问 MessageContext 的所有数据，因此，每一处理器都可以使用、修改请求消息及任何可能存在的响应消息。

配置信息是决定 Handler 调用顺序的一个因素。可以通过修改客户端部署文件的方式来，来向处理链中加入新的 Handler，具体修改如下：

```
〈globalConfiguration〉
  〈requestFlow〉
    〈handler type = "java: org. apache. ws. axis. security. WSDoAllSender"〉
     〈parameter name = "action" value = "Timestamp SAMLTokenUnsigned"/〉
     〈parameter name = "samlPropFile" value = "saml. properties"/〉
    〈/handler〉
  〈/requestFlow〉
〈/globalConfiguration〉
```

其中，WSDoAllSender 类作为一个 Handler 被添加到处理链中，它能够根据 saml. properties 文件中的配置信息向 SOAP 消息的头部嵌入相应的 SAML 声明。

5.2 服务器端

我们在服务器端的处理链中加入一个 Handler 来提取 SOAP 消息中的 SAML 声明。在部署文件中的修改如下：

```
〈requestFlow〉
 〈handler type = "java: handler. SAMLReceiver"〉
 〈parameter name = "action" value = "Timestamp SAMLTokenUnsigned"/〉
 〈/handler〉
〈/requestFlow〉
```

其中，SAMLReceiver 是自定义的类，它从 MessageContext 对象中提取 SAML 声明，获得主体属性。

SAMLReceiver 类同时也作为 ABAC 模型中的策略执行点(PEP)。它收集主体、资源和环境的属性，构成 XACML 授权请求。在实现中，使用了开源工具 sunxacml，它是 XACML 的一个 Java 实现。sumxacml 中用 RequestCtx 类来描述 XACML 授权请求，生成的 RequestCtx 实例被传送到策略决策点(PDP)。

我们编写了一个 SimplePDP 类作为模型中的策略决策

点(PDP)来处理 XACML 请求。系统中采用文件的形式作为策略库,使用 sunxacml 中的 API,策略很容易地被载入 SimplePDP 中。SimplePDP 中的公开接口是

evaluate(RequestCtx request)

SAMLReceiver(PEP)通过调用该接口获得 ResponseCtx 对象。ResponseCtx 类描述了 XACML 响应,SAMLReceiver(PEP)依据 ResponseCtx 对象来决定是否将当前的 SOAP 请求消息传递到实际的 Web 服务。

实验中,我们通过客户端向目标服务发送了多个请求,每个 SOAP 请求中都嵌入了不同的 SAML Token,但只有满足条件的请求才能真正调用 Web 服务,其他请求都被 PEP 拒绝。结果表明,PEP 和 PDP 在服务器端协同合作,能够对 SOAP 请求进行有效的过滤,从而防止相关资源被非法访问。

总结 Web 服务的安全性是制约其发展的关键因素,有效的访问控制机制对 Web 服务至关重要。相比传统模型,基于属性的访问控制(ABAC)具有高度的动态性和灵活性,能够细粒度地进行授权,有效地保护系统资源。同时,该模型与 SAML 和 XACML 标准结合,具有更强的互操作性。

参 考 文 献

1 Wonohoesodo R, Tari Z. A role based access control for Web services. In: IEEE International Conference on Services Computing(SCC 2004), 2004. 49~56
2 Bertino E, Squicciarini A C, Mevi D. A fine-grained access control model for Web services. In: IEEE International Conference on Services Computing(SCC 2004), 2004. 33~40
3 Bhatti R, Bertino E, Ghafoor A. A trust-based context-aware access control model for Web-services. In: IEEE International Conference on Web Services(ICWS'04) Proceedings, 2004. 184~191
4 The Security Assertions Markup Language (SAML) OASIS TC Homepage. http://www.oasisopen.org/committees/tc_home.php? wg_abbrev=security
5 The XML Access Control Markup Language (XACML) OASIS TC Homepage, http://www.oasisopen.org/committees/tc_home.php? wg_abbrev=xacml
6 Web Services Security (WSS) OASIS TC Homepage. http://www.oasisopen.org/committees/tc_home.php? wg_abbrev=wss
7 Axis Architecture Guide. http://ws.apache.org/axis/java/architecture-guide.html
8 Galbraith B, Hankison W, et al. Web 服务安全性高级编程. 北京:清华大学出版, 2003

(上接第 71 页)

簇的合并操作与数据的偏差程度有关,而与新数据到达的数量无关,过多的子簇合并操作会影响到算法的处理速度。图 10 显示了随着数据流的到达算法的处理速度。

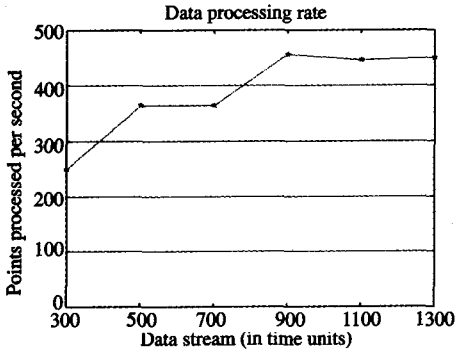

图 10 数据流处理速度

图 10 中可以看出,在开始阶段,由于子簇半径的阈值设置较小,子簇的合并操作较多,处理速度相对较低。随着数据的持续到达,内存中维护的子簇已经能够充分代表网络数据中的绝大部分数据分类,相应地,新到达的数据将被分配到相应的子簇中,此时的操作主要是将新数据加入相应子簇的操作,而非子簇合并操作。因此,当数据流的处理达到一定阶段时,算法的处理速度趋于稳定。

由以上的实验结果可以看出,两阶段的基于数据流的异常入侵检测算法既能够有效地解决网络数据快速到达与检测设备处理能力相对较慢之间的矛盾,又能够充分利用近期的网络数据,更能准确反映当前网络行为的特点,其检测性能优于基于所有历史数据进行入侵检测的算法。

结论 本文提出了一种两阶段的基于数据流的网络异常入侵检测方法,该方法能够在有限的系统资源情况下,以足够快的速度与细粒度保存与维护网络数据的统计信息,并利用该统计信息对不同阶段的网络入侵行为进行检测。理论分析与实验验证的结果显示,这种两阶段的在入侵检测算法,既克服了高速网络环境下内存等系统资源不足对入侵检测算法的影响,又能够准确地反映出现阶段网络数据的行为特征,其检测性能优于基于所有历史数据进行入侵检测的结果,并可以增加系统的灵活性与并行性。同时,算法直接利用未标记的网络数据进行分析计算,也增加了入侵检测方法的实用性。

参 考 文 献

1 Endorf C, Schultz E, Mellander J. Intrusion Detection & Prevention. McGraw-Hill, 2004
2 Lee Wenke, Stolfo S J, Mok K W. A data mining framework for building intrusion detection models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, 1999
3 Cannady J, Mahaffey J. The Application of Artificial Neural Networks to Misuse Detection: Initial Results. In: Proceedings of the 1st International Workshop on Recent Advances in Intrusion Detection (RAID 1998), 1998
4 Mukkamala S, Sung A H. Feature Ranking and Selection for Intrusion detection Systems. In: Proceedings of International Conference on Information and Knowledge Engineering, 2002. 503~509
5 Han Jiawei, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001
6 Breunig M M, Kriegel H P, Ng R T, et al. LOF: Identifying density-based local outliers. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Dallas, 2000. 93~104
7 Portnoy L, Eskin E, Stolfo S J. Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia, 2001
8 Wang Q, Megalooikonomou V. A clustering algorithm for intrusion detection. In: SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005. Orlando, Florida, USA, Mar. 2005
9 Denning D E. An Intrusion Detection Model. IEEE Transaction on Software Engineering, 1987, SE-13: 222~232
10 Babcock B, Babu S, Datar M, et al. Models and issues in data streams. In: Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Madison: ACM Press, 2002. 1~16
11 Ganti V, Gehrke J, Ramakrishnan R. Mining Data Streams Under Block Evolution. SIGKDD Explorations, 2002, 3(2): 1~11
12 Ben-David S, Gehrke J, Kifer D. Detecting Change in Data Streams. In: Proceedings of VLDB, 2004
13 Muthukrishnan S. Data streams: algorithms and applications. In: Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms, 2003
14 Zhang Tian, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, 1996. 103~114
15 KDD99. KDD99 cup dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999
16 Lee Wenke, Stolfo S J. A Framework for Constructing Features and Models for Intrusion Detection Systems. ACM Transactions on Information and System Security, 2000, 3(4): 227~261