语义 Web 服务安全研究*)

杨 欣 沈建京
(信息工程大学理学院 郑州 450001)

摘 要 随着语义 Web 技术的出现和发展,在大规模、开放、异构的分布式环境中实现 Web 服务的自动发现、选择、组合、调用和监控成为可能。语义 Web 服务解决了 Web 服务的语义描述问题,但也面临着与 Web 服务一样的安全、隐私和信任等问题。本文分析了 Web 服务当前面临的主要安全威胁及现有业界安全标准存在的局限性,介绍了安全本体概念并给出了一个基于 OWL 表示的安全本体例子,最后阐述了语义 Web 服务应用安全框架中安全服务的语义描述问题,对 OWL-S 服务本体语言进行扩展,以实现语义安全本体和策略注释。

关键词 Web 服务,语义 Web 服务,安全本体,安全服务

Research on Semantic Web Services Security

YANG Xin SHEN Jian-Jing
(Institute of Science, Information Engineering University, Zhengzhou 450001)

Abstract The emergence and development of Semantic Web technology facilitate the higher automation of Web service discovery, selection, composition, invocation and monitoring in a large-scale, open, heterogeneous, and distributed environment. Although the Semantic Web services addresses the description of the semantics of Web services, it also faces the challenge of problems including security, privacy and trust like Web services. The main security threats of Web services and the limitation of current security-related standards are analyzed in this paper. Security ontology is presented and the definition of a security ontology in OWL is illustrated. Finally, the description of the semantics of security services included in a security framework for Semantic Web Services applications is detailed by extending the OWL-S profile to make security ontology and policy annotations.

Keywords Web services, Semantic Web services, Security ontology, Security services

1 引言

Web Services(WS)被普遍认为是新一代应用程序集成以及通向新的商业模式的大门,是企业之间相互联系的前所未有的重要途径。WS 利用基于 XML 的开放标准(如 SOAP、WSDL 和 UDDI)和广泛使用的 Internet 传输协议(如 HTTP)实现了面向服务的体系结构(SOA),因此特别适合于连接跨企业边界的业务过程[1]。然而,随着 Web 服务的应用和发展,越来越多的问题逐渐暴露出来。一方面,WS 的应用范围从 Extranet 推广到 Internet,使得更多的数据和应用暴露在黑客和信息窃贼的视野之内,企业信息面临的安全威胁也随之显著增加,从而严重地阻碍了 WS 的广泛应用。作为一种新兴的分布式计算技术,WS 不仅继承了分布式计算固有的安全问题,还面临着由于其自身的动态性等特点所带来的新的安全威胁,所以传统的安全机制(如加密、数字签名、PKI 等)根本无法为 WS 提供充分的安全保证。同时,当前的工业标准也无法在抽象层次上很好地解决 WS 安全、隐私和信任问题。另一方面,虽然 WS 提供了这种分布式异构系统之间的互操作性,由于相应的规范都缺乏语义支持,计算机无法识别和理解每个 Web 页面的内容,使得复杂 Web 服务应用需要更多的人为参与来实现,从而无法有效而灵活地支持查找、分配、组合、运行时监控或调用 WS。

互联网的创始人 Tim Berners-Lee 在 1998 年提出了在现有 Web 的基础上建设下一代 Web 的蓝图-Semantic Web (SW)[2],即各种资源被人为地赋予了各种明确的语义信息,计算机可以分辨和识别这些语义信息,并对其自动进行解释、交换和处理。SW 能够以计算机可读和可理解的方式对 Web 数据进行注释,使得机器之间处理和交换基于 Internet 的信息比当前基于 XML 的允许互操作的方法更加有效和有意义。

在 SW 技术的推动下,为了弥补 WS 缺乏语义描述这个不足,Semantic Web Services(SWS)诞生了。作为一个新的研究方向,SWS 以计算机可解释的语言,采用丰富的语义注释,为实现自动化的 WS 发现、调用、组合、监控和恢复提供了有效的解决方案。丰富的语义能够提供更加充分而灵活的服务交互自动化,能够支持构建功能更加强大的工具和相关的方法学[3]。然而,SWS 并不是为解决 WS 安全问题而提出的。与 WS 一样,SWS 也面临着安全、隐私和信任等问题。为了提供一个安全可信的在线交易环境,建立一个灵活的、自适应的 SWS 应用安全框架是我们的研究目标。将与基于 Web 的应用、Web 资源或服务的安全方面有关的语义注释,与支持决策过程(比如决定谁有权访问什么样的资源或者根据给定的安全约束选择适当的服务)的推理系统结合在一起可以很好地推动 WS 安全乃至 SWS 安全的实现。

本文的内容安排如下:第 2 节分析了 WS 当前面临的主要安全威胁及现有业界安全标准存在的局限性。第 3 节介绍

*)河南省自然科学基金(编号 0511011300)。杨 欣 讲师,博士研究生;沈建京 教授,博士。

了安全本体概念并给出了一个基于 OWL 表示的安全本体例子。第 4 节重点阐述了 SWS 应用安全框架中安全服务的语义描述问题，采用 OWL-S 服务本体语言实现语义安全本体和策略注释。最后是结束语和下一步工作展望。

2 WS 安全现状

基于这种分布式、分散的体系结构，WS 在为预先确定的或未知的客户端提供定义明确的服务的同时，还必须解决通信实体之间在通信过程中所面临的典型安全问题，包括鉴别、授权、机密性、数据完整性、不可抵赖性和可用性。由于 WS 自身具有的开放性、动态性和互操作性等特点，WS 所面临的安全威胁不同于传统的基于 Web 的应用所面临的安全威胁，一些新的安全威胁正困扰着 WS 的开发者和使用者。例如，恶意内容攻击试图强制一个 SOAP 节点(服务器)去做其它的事情，比如检索未被授权访问的数据，或者通过 SQL 注入来破坏数据和操纵 SOAP 消息中的内容，从而导致消息接收方消耗大量资源、崩溃或者无法响应。据统计，近年来排在前 10 位的 WS/XML 安全威胁来自于[4]：

(1)强制分析，发送有效的但是恶意的 XML 文档到 SOAP/XML 节点，从而导致占用大量资源；

(2)外部实体攻击，将递归元素、超大载荷或者 XQuery/SQL 命令注入到一个有效文档中；

(3)超大载荷，迫使 XML 分析器检查超大型 XML 文档来占用系统资源(内存和 CPU)；

(4)恶意变形，将 XML 文档处理成为一种并非设计者本意的形式；

(5)递归元素，一个相对较小的 XML 文档的一部分，但却能够容易地被扩展成超大字节的 XML 文档，从而迫使 XML 分析器占用所有可用的资源；

(6)路由迂回，将路由信息插入到一个 SOAP 消息中，迫使数据流在到达最终目的地前通过中介体被路由到其它地方；

(7)Schema 感染，修改一个外部引用的 Schema 文档；

(8)WSDL 列举，一个未授权的客户端通过读取 WSDL 来获得关于一个部署 WS 的组织的操作和 Schema 信息；

(9)XML 封装，使用 CDATA 标记将系统命令嵌入到 XML 载荷中来获得远程访问能力(CDATA 标记用于实现向下兼容性)；

(10)XQuery 注入，强制 SOAP 节点对一个 XML 文档执行其它功能。

当前，针对上述这些威胁，低层的加密、数字签名机制、证书和公钥基础设施只能为基于 Web 的交互提供一个良好的安全基础设施。然而，提供高层的安全性，特别是在无法预先建立信任关系的动态交互情形下，需要依赖于多种特定机制，这些机制间的异构性却带来了严重的安全漏洞和不良后果。目前已提出的与安全有关的工业标准都是建立在假定 B2B 实体间存在信任关系的基础上。例如，由 IBM、Microsoft 和 VeriSign 联合提出的 WS-Security[5]，WS-Trust[6]和 WS-Policy[7]、OASIS 安全服务技术委员会提出的 SAML(Security Assertion Markup Language)[8]，Sun Microsystems 倡导的 Liberty Alliance Project 所开发的一组安全规范。WS-Security 提供了在 SOAP 之上的安全层，描述了如何将签名和加密信息或者安全令牌附加到 SOAP 消息中。WS-Trust 描述了现有的直接信任关系如何通过创建安全令牌发布服务来实现间接的代理信任。WS-Policy 提供了描述和传播 Web 服务策略的通用模型和语法，使得 WS 能够根据它们的能力、需求和其它特征来描述策略。SAML 用于在不同的 Internet 安全域中交换身份验证和授权凭证。

上述这些标准支持的是与凭证格式或编码字符集有关的低层的安全或策略标注，没有解决语义上的特定于用户或应用程序的信任令牌及其关系，也没有考虑到策略的表示性问题。这些标准只是满足了那些在操作和交互之前就已经建立了可信的伙伴和业务关系的 B2B 应用需求。如今越来越多的开放和内部服务都可以通过 Internet 获得，在这样的环境中预先建立通信实体间的信任关系已不再现实。跨不同领域和应用平台的实体彼此间需要进行互操作，简单的基于身份或角色等鉴别机制显然也已经无法满足这种需求。显而易见，这些标准无法扩展到这种动态的环境中，因此必须在传统安全机制的基础上采取其它途径来解决 WS 面临的安全问题。

3 安全本体

在一个 Agents 和 WSs 都有安全需求并能提供安全能力的交互情形中，一个 Agent 希望找到一个能够实现特定功能并满足特定安全需求(比如不需要鉴别而使用加密消息进行通信)的 WS，Agent 本身也具有一定的安全能力(比如拥有一些凭证或者能够使用一些安全协议)。同样，一个 WS 在提供一定的安全能力(比如所使用的安全机制和能够接受的凭证等)的同时也拥有自己的安全需求(即希望与具备什么样安全能力的 Agent 交互)。因此，Agent 在查找所需的 WS 时，即使找到功能匹配的服务，也会因为安全需求和安全能力的不匹配而无法找到合适的 WS[9]。

安全本体提供了一种表示这种安全需求和安全能力的方法，在现有安全标准之上的更高抽象层次上总结了一些常用的与安全相关的标记。这些标记可以用来对 Web 资源、服务和 Agent 进行安全注释。同时，这些安全本体也为实现自动的基于安全注释的假定推理奠定了基础。

OWL(Web Ontology Language)[10]是一个建立在 RDF 基础上的本体语言，对 RDF schema 进行了扩展，提供了丰富的词汇来表示复杂关系和提高推理能力。OWL 支持类定义和子类(继承)、属性定义和属性分层、基数限制等概念，因此，OWL 非常适合于描述安全本体。

当前，DAML Project 下的 DAML-S 安全工作小组已经初步开发了一些安全本体[11]。例如，作为调用那些常用安全标准和协议的高层抽象接口，安全机制本体 SecurityMechanism 的 OWL 定义如下(详见文[12])：

```
//顶层类
〈owl:Class rdf:ID="SecurityMechanism"〉
〈/owl:Class〉
//特性，包括 documentation、syntax、relSecNotation、enc、
  sig、reqCredential
〈owl:ObjectProperty rdf:ID="syntax"〉
  〈rdfs: domain rdf: resource = "# SecurityMecha-
  nism"/〉
  〈rdfs:range rdf:resource="#Syntax"/〉
〈/owl:ObjectProperty〉

〈owl:ObjectProperty rdf:ID="relSecNotation"〉
```

```
〈rdfs: domain rdf: resource = "# SecurityMechanism"/〉
〈rdfs:range rdf:resource="#SecurityNotation"/〉
〈/owl:ObjectProperty〉
……
//SecurityMechanism类的所有子类,包括 Syntax、SecurityNotation、KeyFormat、Encryption、Signature、Protocol、DataTransferProtocol、KeyProtocol、KeyRegistrationProt、KeyInformationProt
〈owl:Class rdf:ID="Syntax"〉
〈rdfs: subClassOf rdf: resource = "# SecurityMechanism"/〉
〈/owl:Class〉

〈owl:Class rdf:ID="SecurityNotation"〉
〈rdfs: subClassOf rdf: resource = "# SecurityMechanism"/〉
〈/owl:Class〉
……
```

//各个子类及限制类(用于定义特定的安全特征)的实例,包括 Syntax、SecurityNotation、KeyFormat、Encryption、Signature、PolicyLanguage、Protocol 及拥有特定 syntax、relSecurityNotations 和 reqCredentials 的限制类

```
〈Syntax rdf:ID="ASCII"〉
〈/Syntax〉
……
〈SecurityNotation rdf:ID="Authentication"〉
〈rdfs:comment〉
    Mechanism to decide whether a given credential
    (key, certificate, etc.)stands for an entity.
〈/rdfs:comment〉
〈/SecurityNotation〉
……
```

每一个子类可以对应很多实例。比如 syntax 类的实例包括 ASCII、OWL、RDF、XML、MIME;SecurityNotation 类的实例包括 Authentication、Authorization、AccessControl、DataIntegrity、Confidentiality、Privacy、ExposureControl、Anonymity、Negotiation、Policy 和 KeyDistribution;X.509 可以是 KeyFormat 类的实例;XML-DSIG 可以是 Signature 类的实例;OpenPGP-Enc 可以是 Encryption 类的实例,等等。

4 SWS 安全

为了解决 WS 的语义描述问题,实现 WS 的自动发现、调用、组合、监控和恢复,SWS 解决方案随着 SW 技术的提出应运而生。然而,由于 WS 是将跨应用平台和组织边界的各个独立的控制域连接在一起,服务保护和安全、结果的可靠性及来源验证等相关问题并不能因 WS 的自动化实现而得到有效解决。为了营造一个 SWS 应用的安全环境,需要建立一个包括像安全服务、鉴别和授权协议,及用于交换和协商策略的技术等在内的 SWS 应用安全框架,以支持在客户端和 WS 之间(或 WS 和 WS 之间)建立信任关系。

当前,描述和部署安全服务是实现这个目标的第一步。安全服务是建立一个可信在线交易环境所必不可少的一个组成部分,也是建立 SWS 应用安全框架的基础。在语义注释的帮助下,安全服务能够提供定义明确的功能,比如签名或验证 XML 签名或加密和解密消息和文件。安全服务也可以组合以实现更为复杂的安全机制。为了满足各种应用程序的安全需求,可以重用安全服务而不必为每个 WS 应用实现新的安全机制。安全服务就是以 WS 的形式提供具有语义意义的、说明性的、计算机可处理的安全能力描述。这样,在动态产生组合 WS 应用的过程中,服务匹配器(matchmaker)就能够发现这些定义明确的高层安全 WS 并使用它们以满足服务请求者和提供者的安全需求。而且,应用程序开发者和工程人员也不必在创建 WS 应用过程中过多考虑与安全有关的问题。

为了实现语义描述安全服务,OWL-S 是为安全本体和 WS 之间建立连接的一个有效解决办法。OWL-S(OWL Services)[13]以明确的计算机可解释的形式,为 WS 提供者提供了一组用来描述 WS 特性和能力的基于 OWL 的标注语言构造符。WS 的 OWL-S 标注可以支持 WS 发现、执行、互操作、组合和执行监控的自动化实现。一个 OWL-S 描述包含一个 Profile(提供 WS 的一般性描述)、一个 Process Model(描述 WS 如何执行任务和 WS 交互协议)和一个 Grounding(说明 Process Model 中的原子过程如何映射到 WSDL 表示)。OWL-S 本身并不支持安全注释和策略,所以必须要对 OWL-S 的 Profile 进行扩展。

4.1 安全本体注释

将前文描述的安全本体 SecurityMechanism 类定义为 ServiceParameter 类的子类,并声明两个新的特性:securityRequirement 和 securityCapability。值域均为 SecurityMechanism 类。利用这两个特性,WS 和 Agent 各自的安全需求和能力可以得到很好的描述。一个 Agent,请求一个能够执行鉴别的 WS,其 Profile 描述为:

```
〈security:KeyProtocol rdf:ID="Sec2"〉
〈security: relSecNotation rdf: resource = "&security;
#Authentication" /〉
〈/security:KeyProtocol〉
〈profile:Profile rdf:ID="RequestServiceProfile1"〉
〈profile:serviceName〉…〈/profile:serviceName〉
〈profile:textDescription〉…〈/profile:textDescription〉
〈sec-requirement rdf:resource="#Sec2"/〉
〈/profile:Profile〉
```

一个声称能够使用 XKMS 协议进行通信的 WS 的 Profile 描述如下:

```
〈security:XKMS rdf:ID="Sec1"/〉
〈profile:Profile rdf:ID="RegisteredServiceProfile1"〉
…
〈sec-capability rdf:resource="Sec1"/〉
〈/profile:Profile〉
```

WS 的安全能力是否满足 Agent 的安全需求取决于安全推理器。安全推理器实现了处理安全信息匹配的推理算法,比如像 Java 的 JTP(Java Theorem Prover)。

采用安全本体方法对 OWL-S 进行扩展,实现了在较高的抽象层次上对 WS 和 Agent 安全需求和能力的全局描述,为安全服务的查找和匹配奠定了基础。然而,这种本体描述是静态的,是按照事先定义的安全类进行描述的,而不是根据动作来描述的。为了弥补这个不足,可以采用安全策略描述来增强安全服务的语义描述。

4.2 策略注释

策略描述的是WS的一般特性而不是专属于某个过程的特性，所以可以将策略描述与Profile结合在一起。利用SWS提供的丰富的语义注释，通过提供具有语义意义的、说明性的、计算机可处理的策略描述和相关的算法和工具来实现服务请求者与服务提供者之间的策略匹配，从而解决SWS面临的安全问题。

将策略集成到SWS描述中可以提供多层次的安全性。策略可以限制谁能够在何种条件下使用一项服务，相关信息应当如何提交给服务以及服务要如何使用所提供的信息。策略说明就是对请求者、服务和内容不同属性所实施的约束。例如，授权策略，定义了一组限制访问服务的规则，只有特定的用户在特定的环境下才能访问服务；隐私策略，描述了什么样的信息能够被交换，信息的合法使用以及信息交换应当处于什么样的条件下；机密性策略，描述了一个服务的输入和输出参数的密码特征。

为了实现在服务的安全需求中包含策略说明，只须增加一个新的特性作为securityRequirement的子特性，用来描述为正确执行服务所必须实施的各种策略。

为了实现一个SWS应用安全框架，除了提供可部署的、具有明确形式定义OWL-S接口的安全服务外，还需要部署各种会话、鉴别或交互协议以支持密钥或策略的建立和交换及协约的协商。而且，框架中的服务和协议都需要进行语义注释以为那些执行服务分析和组合的语义工具所使用。

结束语 在SW技术的推动下，SWS近年来受到了学术界的广泛关注。SWS是在WS中加入了语义的支持，使得计算机之间能够理解互相通信的内容，从而实现WS的自动发现、调用和组合。SWS是为解决WS的语义描述问题而提出的，面临着与WS一样的安全、隐私和信任等问题。因此，营造一个SWS应用安全环境，既是为了满足下一代WWW应用的发展需要，也为复杂的SWS应用实现安全可信的在线交易提供了必要保证。

提出并建立一个灵活的、自适应的SWS应用安全框架是我们的最终研究目标。当前，DAML Project下的DAML-S安全工作小组已经开发了一些与安全有关的本体，比如凭证本体、安全机制本体、服务安全扩展本体、Agent安全扩展本体及隐私本体等。在安全本体研究的基础上，安全服务的语义描述、实现和部署是实现这个目标所需要完成的第一步。本文提出了采用OWL-S安全本体和策略注释的方法来解决安全服务的语义描述问题。下一步的工作就是考虑如何实现和部署安全服务。如何实现安全推理器中的安全匹配算法，采用哪一种策略表示语言，选用微软的.Net平台还是Sun的J2EE标准，采用两层还是三层部署结构等一些问题还有待进一步解决。

参 考 文 献

1 Kearney P. Message Level Security for Web Services. Information Security Technical Report, 2005,10:41～50
2 Lee TB, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001,5:34～43
3 Denker G, Kagal L, Finin T. Security in the Semantic Web Using OWL. Information Security Technical Report, 2005,10:51～58
4 MacVittie L. Web Services Security. NetworkComputing. http://www.networkcomputing.com/showArticle.jhtml?articleID=21400998&pgno=3, 2004
5 Atkinson B, Della-Libera G, Hada S, et al. Web services security(WS-Security). Http://www-128.ibm.com/developerworks/Webservices/library/ws-secure/, 2002
6 Anderson S, Bohren J, Boubez T, et al. Web Services Trust Language(WS-Trust). ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf, 2005
7 Bajaj S, Box D, Chappell D, et al. Web Services Policy Framework(WS-Policy). http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-polfram/ws-policy-2006-03-01.pdf, 2006
8 SAML. Assertions and Protocols for the OASIS Security Assertion Markup Language(SAML) V2.0. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 2005
9 Denker G, Kagal L, Finin T. Security for DAML Web Services: Annotation and Matchmaking. In:Proceedings of the 2nd International Semantic Web Conference (ISWC2003), LNCS2870, Springer-Verlag, 2003.335～350
10 Web Ontology Working Group. OWL. Http://www.w3.org/2001/sw/WebOnt/
11 DAML Services - Security and Privacy. Http://www.daml.org/services/owl-s/security.html
12 http://www.daml.org/services/owl-s/security/security.owl
13 DAML Services. Http://www.daml.org/services/owl-s

(上接第81页)

参 考 文 献

1 Chaum D, Heyst V E. Group Signatures[C]. In:Proc. of EUROCRYPT'91. Lecture Notes in Computer Science, 1991, 547:257～265
2 Chen L, Pedersen T. New group signature schemes[A]. In:Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science. Springer-Verlag, 1995, 950: 171～1811
3 Camenish J, Stadler M. Efficient group signatures for large groups[C]. In:Proc. of CRYPTO'97. Lecture Notes in Computer Science, 1997, 1296: 410～424
4 Camenisch J. Group signature schemes and payment systems based on the discrete logarithm problem:[PhD thesis]. vol. 2 of ETH Series in Information Security and Cryptography, ISBN 3-89649-286-1. Hartung-Gorre Verlag, Konstanz, 1998
5 Ateniese G, Tsudik G. Some open issues and new directions in group signatures[C]. In: Franklin M, ed. Financial Cryptography Conf. LNCS 1648, Berlin: Springer-Verlag, 1999. 196～211
6 Ateniese G, Camenish J, Joye M, et al. A practical and provably secure coalition-resistant group signature scheme [A]. In: M. Bellare, ed. Advances in Crypto'2000, Berlin: Springer-Verlag, 2000. 255～270
7 Ateniese G, Medeiros B. Efficient Group Signatures without Trapdoors. In: Advances in Cryptology - ASIACRYPT'2003, LNCS2894, 2003. 246～268
8 Boneh D, Boyen X, Shacham H. Short group signatures. In: CRYPTO'2004, LNCS. Springer Verlag, 2004
9 Camenisch J, Groth J. Group Signatures: Better Efficiency and New Theoretical Aspects. In: SCN 2004, LNCS 3352, 2005. 120～133
10 Fiat A, Shamir A. How to prove yourself: practical solutions to identification and signature problems. In: Advances in Cryptology - CRYPTO'86, Springer-Verlag, LNCS263, 1987. 186～194
11 Camenisch J, Stadler M. Proof systems for general statements about discrete logarithms:[Technical Report]. TR 260, Institute for Theoretical Computer Science, ETH Zurich, Mar. 1997
12 Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In: First ACM Conf. on Computer and Communications Security, New York: ACM Press, 1993. 62～73
13 Schnorr C P. Efficient identification and signature generation for smart cards. In: CRYPT'89, LNCS435, Springer-Verlag, 1990. 239～252
14 Pointcheval D, Stern J. Security proofs for signature schemes. In:Cryptology-Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science 1070, Springer-Verlag, 1996. 387～398
15 张键红，伍前红，邹建成，王育民. 一种高效的群签名，2005,33(6):1113～1115