第33卷 第22期 计 算 机 工 程 2007年11月
Vol.33 No.22 Computer Engineering November 2007
·安全技术· 文章编号：1000—3428(2007)22—0190—03 文献标识码：A 中图分类号：TP309

Web Service的安全机制

钱 权，严家德
(上海大学计算机工程与科学学院，上海 200072)

摘 要：Web Service采用松散的方式将计算服务整合在一起，在电子商务、企业应用系统集成等分布式计算环境中发挥着重要的作用，随着Web Service应用的普及，安全问题也受到了重视。针对利用SSL和防火墙技术实现Web Service安全的不足，本文从Web Service的体系结构入手，将Web Service的安全分为企业处理层安全、Web Service目录及注册层安全、通信层安全3个层次，并阐明了Web Service不同层次的安全策略和实现方法。

关键词：Web Service；安全机制；SOAP 安全；UDDI安全

Security Mechanisms of Web Service

QIAN Quan, YAN Jia-de
(School of Computer Engineering & Science, Shanghai University, Shanghai 200072)

【Abstract】Web Service plays an important role in integration of E-Commerce and business application systems in distributed computing environments, which loosely couple the computation services across network. With the development of Web Service in different areas, security issues are being widely focused on. Trough analyzing the Web Service architecture and implementation deficiency of Web service security only by SSL or firewall, this paper divides the Web service security into three levels: business process level, Web service catalog and registry level, and communication level. The different security strategies used in different levels and the corresponding implementation methods are also discussed in detail in the paper.

【Key words】Web Service; security mechanisms, SOAP security; UDDI security

目前，Web Service在分布式计算领域中发挥着重要的作用。Web Service将计算服务以松散的方式整合在一起，可应用于电子商务、应用系统集成中。Web Service可以提供实时的交互，这种交互主要是应用与应用的交互，而不是人与应用的交互。

另外，Web Service是一种较典型的SOA(service oriented architecture)结构，SOA提供一组松耦合的服务，每一个服务的建立和替换都是相对“低廉”的。与传统的紧耦合架构相比，松耦合架构更能适应业务的变化。在SOA中，可以用一个服务替换另外一个服务，而不需要关心底层的实现技术。采用SOA可以充分利用企业现有的IT资产，包括遗留应用和数据库，新系统可以将遗留应用和数据纳入SOA而不会替换它们。采用SOA可以使企业的IT架构能够快速、有效地适应业务需求的变化。通过使用Web service，可以将企业应用框架转变成SOA，帮助企业实现快速且高效的应用。

1 Web Service安全机制分析

1.1 SSL的缺陷

目前，已有的安全标准SSL，它是一种较好的提供安全电子商务交易的机制。但为什么不能直接使用SSL来提供安全的Web Service呢？事实上，Web Service需要end-to -end的安全，然而SSL提供的是P2P的安全。传统的电子商务大多在HTTP协议上通过SSL,PKI以及防火墙来提供安全。SSL是在Web浏览器和Web服务器之间(P2P)建立安全的连接，其提供的安全特性包括：认证(提供浏览器和服务器间的认证)，保密(提供请求和响应之间数据的加密)和完整性(保证在请求和响应的数据在传输过程中，数据不会被修改)。SSL应用于Web service的不足之处有：

(1)SSL提供P2P安全，而Web Service需要end-to-end安全，此时端到端之间可能还有众多的中间节点，中间节点之间存在大量的基于XML的信息交互，SSL无法为其提供安全服务。

(2)SSL在传输级提供安全，而Web Service需要“消息级”的安全。使用SSL可以保证消息在传输过程中的安全性，但无法保证消息的后续安全性。此外，由于SSL只能提供传输级安全性，因此无法实现交互中XML文档在元素级别的签名和加密。

(3)SSL不能提供交易的不可抵赖性。SSL不能提供端到端的服务请求和响应交易的不可抵赖性。

1.2 基于防火墙的Web Service安全

利用防火墙提供Web Service安全。由于Web service属于企业的核心应用，不能将其部署在防火墙的非军事化区(demilitarized zone, DMZ)，DMZ区在企业的内部网之外，为外部用户提供公共访问服务。此时的SOAP消息可能包含恶意数据，并破坏企业应用安全，因此，需要防火墙对SOAP消息进行细致的内容审查和过滤。防火墙的审查内容包括：

(1)SOAP请求消息是不是发往一个处于活动状态的Web

基金项目：上海市教委科技发展基金资助项目(205649)；优秀青年基金资助项目(99-0303-06030)
作者简介：钱 权(1972－)，男，博士，主研方向：网络安全，人工智能；严家德，硕士研究生
收稿日期：2006-12-03 **E-mail：**qqian@staff.shu.edu.cn

Service。

(2)判断 SOAP 请求和 SOAP 消息的合法性。

(3)判断 SOAP 消息中数据的合法性。

基于防火墙的 Web Service 安全的不足之处有：

(1)在目前已有的防火墙中，很难做到对 SOAP 消息详细的内容审查，需要进行专门的研发工作。

(2)Web Service 部署在企业内部网中，如果 Web Service 的安全受到威胁，整个企业内部网的安全将会受到威胁。

(3)从 Web Service 的应用类型看，存在简单 RPC 代理方式、联邦方式和事务方式，不同的应用方式，对“安全需求”的要求和复杂性是不一样，采用防火墙集中对访问进行审查，其控制规则过于复杂，很难保证防火墙的处理效率。

(4)若采用不同的防火墙来处理不同安全级别的 Web Service，每一个防火墙针对其安全需求制定相应的安全策略。这种方式处理简单，但是缺点也是非常明显的。因为在分布式处理环境中，Web Service 的应用是非常复杂的，可以提供给 Web 用户使用，也可以提供作为联邦处理链中的一环给其他 Web Service 调用，所以每一种 Web service 需要一个防火墙保护，代价太大，且没有办法为每一个 Web Service 制定一种安全策略。

2 Web Service 层次化安全设计

2.1 Web Service 的体系结构

Web Service 的体系结构从上向下可以分为企业业务逻辑处理层、Web Service 的目录分类和注册层、通信层。其体系结构如图 1 所示。

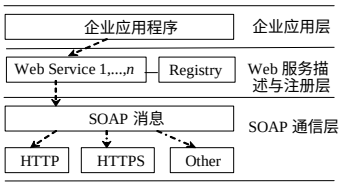

图 1 Web Service 体系结构

2.2 Web Service 层次安全设计

针对 Web Service 的体系结构，可以将其安全性按层次划分为：(1)企业处理层安全；(2)Web Service 分类和注册层安全；(3)通信层安全。

2.2.1 企业处理层安全

企业处理过程(business process)的集成和自动化是企业分布式应用首要解决的问题。已有的技术如CORBA, JMS, BPMS等在提供跨企业应用集成中有着处理代价高、难于实现等缺点。而采用基于SOA的架构，企业处理过程采用基于XML的规范化的描述能与Web Service相结合，提供无缝的企业分布式处理集成。目前企业处理流程标准有ebXML, BPSS, XLANG, WSFL, BPML等[1]。企业处理层的安全主要依靠企业处理流程的安全措施，目前这方面的研究还不多，众多企业处理标准在安全方面也很少涉及。

2.2.2 Web Service 分类及注册层安全

Web Service是一种典型的SOA结构，由于在SOA体系结构中服务是采用松耦合的方式，因此服务间的依赖关系必须在运行时刻依照某种发现机制加以解决。Web Service中这种发现机制有统一描述发现集成协议(universal description discovery and integration, UDDI)[2]和ID-WSF[3]发现方法。常用的是UDDI机制。

UDDI的安全包括UDDI注册表的安全、UDDI交易安全、UDDI基础设施安全。UDDI 注册表安全主要包括：

(1)内容认证：注册表中的内容的确是已知的注册表的所有者或维护者所放置的。

(2)内容授权：UDDI 注册表提供给 UDDI 用户使用的数据是经过经过签名的，并经过授权的合法用户提交的。

(3)内容未被修改：从 UDDI 注册表中获取 Web 服务时，内容是未修改过的。

(4)内容的新鲜性：保证 Web 服务的提供者放在 UDDI 注册表中的内容是最新的。

(5)内容的秘密性：UDDI 数据在交换过程中，保证交换内容的秘密性。

UDDI 的交易安全是保证在 Web 服务的提供者和消费者之间的事务交易以一种可信任的方式运行，信任参数为交易的双方所知道，且这种信任关系在服务的调用和执行过程中始终存在。

UDDI 基础设施安全是保证 Web 服务的提供者和消费者在交易时所有基础设施环境的安全，包括 Web 服务的发现、服务的调用和执行的安全性。

下面将研究一种基于访问控制的XACML[4]实现UDDI安全的方法。XACML由OASIS制定，帮助用户在XML中定义其授权和访问策略。XACML包括 3 个主要的元素：Resource, Subject, Action。在XACML中Web服务的提供者和消费者指派给Subject；UDDI指派给Resource；对资源的访问授权指派给Action。基于XACML实现对UDDI的访问控制的过程如图 2 所示。

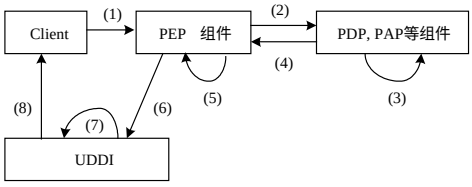

图 2 基于 XACML 的 UDDI 访问控制流程

(1)经过授权的用户向 XACML 的 PEP(policy enforcement point)组件请求 UDDI 资源。

(2)PEP 组件转发访问请求给 XACML 的 PDP,PAP 等组件。

(3)XACML 的 PDP,PAP 等组件验证用户的策略。

(4)包含特定访问策略上下文的应答传送给 PEP。

(5)应答经过处理后，用户的访问许可被确定。

(6)如果用户的请求动作被允许，则将请求传送给 UDDI。

(7)UDDI 处理用户的请求动作。

(8)请求的结果返回给用户。

将 XACML 应用到 UDDI，需要用到的 XACML 属性有：

(1)资源绑定(resource binding)：资源绑定是对资源的描述包括资源的名称、内容、所有者等属性加以描述。

(2)动作绑定(action binding)：XACML 中的动作绑定是 UDDI 所提供的一系列请求和发布 API，请求 API 是对注册表中 business,service,binding,tModel 等提供的查询(find)和详细查询(get details)。发布 API 是保存、删除、认证令牌操作。

(3)主体绑定(subject binding)：主体绑定是约束特定主体对主体的访问，描述主体的属性包括主体的 ID、角色(role)和隶属的组(group)。

使用 XACML 来约束用户对 UDDI 中 Web Service 的访问控制，需要制定访问控制规则，在 XACML 中也称为控制

策略。UDDI 注册表常见的控制策略有：

(1)任何用户都可以检索注册在 UDDI 注册表中的 Business 信息。

(2)只有经过授权的个人才能发布或改变注册表中的信息；

(3)只有信息的原始提供者才能进行信息的更改或删除；

(4)每一个 UDDI 注册表的实例能自定义其用户认证机制。

例如使用 XACML 实现上述 Web Service，其提供者可以进行信息的更改和删除，XACML 访问策略应用如下：

```
<Policy PolicyID="urn:oasis:names:tc:webservice-uddi:3.0:plicy:policyid:permit-serviceProvider"
RuleCombiningAlgID="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Rule RuleID="urn:oasis:names:tc:webservice-uddi:3.0:example:ruleid:2" Effect="Permit">
    <Description>Web Service 的提供者可以进行信息的更改和删除 </description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchID="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#String">
                Service Provider
            </AttributeValue>
          </SubjectMatch>
        </Subject>
      <Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchID="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                business_entity</AttributeValue>
            <ResourceAttributeDesignator AttributeID="oasis:names:tc:webservice-uddi:3.0:resource"

DataType="http://www.w3.org/2001/XMLSchema#string" />
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchID="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                delete_business</AttributeValue>
            <ActionAttributeDesignator AttributeID="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string" />
          <ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Conditon FunctionID=... />
  </Rule>
</Policy>
```

2.2.3 通信层安全

Web Service 通常采用 SOAP(simple object access protocol) 协议作为通信层协议。SOAP 消息是由 SOAP envelope、可选的SOAP header和SOAP body组成的XML文档[5]。SOAP消息的结构如图 3 所示。

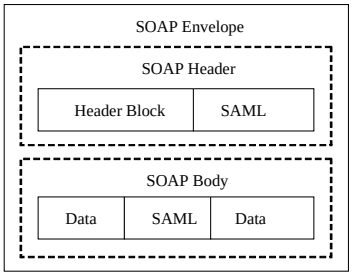

图 3 SOAP 消息结构

SOAP 协议本身并没有定义安全性，所有的 SOAP 安全性依赖于其安全性的扩展。目前 W3C 的 XML 加密工作组进行定义，标准包括 XML 文档的加密、解密等。XML 的加密主要用于对 XML 数据保密性，此外利用公钥基础设施 PKI 能够提供 XML 数据的认证、数字签名以及密钥交换。基于 XML 的 PKI 应用被大大简化，XKMS(XML key management specification) 主要是为了实现 PKI 和数字证书功能。

3 结束语

作为一个 SOA 架构，Web Service 采用服务松散耦合的方式，为企业内部或企业之间应用系统的信息交互，提供快速且低廉的解决方案。

目前，针对 Web Service，存在众多的安全规范，如企业处理流程规范、XML 安全规范、SOAP 安全规范、WS 系列安全规范等，如何高效且合理地运用这些规范，是需要解决的问题。面临的问题有：

(1)Web Service 作为基础的分布式计算环境，其应用环境、应用方式、安全需求和安全级别，各不相同。

(2)安全措施和处理效率之间存在一定矛盾。

如何在保证安全的前提下，不增加额外的安全处理开销，仍需要进一步的研究。

参考文献

1 Clark M, Fletcher P. Web Service Business Strategies and Architectures[M]. Birmingham, UK: Wrox Press, 2002-08.

2 UDDI ORG. UDDI Version3.0 Specifications[EB/OL]. (2004-09). http://uddi. org/pubs/uddi-v3.00-published-20020719.htm.

3 Sergent J. Liberty ID-WSF Discovery Protocol[EB/OL]. (2003-08). http://www.projectliberty.org/liberty-idwsf-disco-svc-v1.1.pdf.

4 OASIS. OASIS Extensible Access Control Markup Language [EB/OL]. (2002-07-12). http://www.oasis-open.org/committees/xacml/repository/cs-xacml-core-01.pdf.

5 W3C. SOAP Version 1.2 Specification, W3C Working Draft [EB/OL]. (2001-07-09). http://www.w3.org/TR/2001/WD-soap12-20010709/.