一种基于SOA的SOAP消息安全传输机制

华 悦[1] 徐 涛[1,2]
(南京航空天航大学计算机科学与技术学院 南京 210016)[1]
(中国民航大学计算机科学与技术学院 天津 300300)[2]

摘 要 随着SOA技术的发展与普及应用,基于SOA的Web服务安全问题日益突出,而SOAP消息传输的安全性是决定Web服务安全的重要因素。目前SOAP消息的传输主要依赖于WS安全标准,但由于WS安全标准存在种种缺陷,因此SOAP消息在传输过程中会受到XML注入攻击等Web攻击。提出了一种新的SOAP消息安全传输机制,即在现有的基于WS安全标准的安全传输机制基础上添加SOAP Validation节点。最后通过实验验证,该安全传输机制能检测出XML注入攻击,提高SOAP消息传输的安全性。

关键词 SOA,SOAP,安全传输机制,XML注入攻击
中图法分类号 TP309 **文献标识码** A

SOAP Message Security Transport Mechanism Based on SOA

HUA Yue[1] XU Tao[1,2]
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[1]
(College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)[2]

Abstract With the technology development and popularization of SOA applications, the security issues of Web services based on SOA have become increasingly prominent. The security of SOAP message is of great importance to Web service security. Currently SOAP message transport mainly depends on the WS-Security standards. However, the WS-Security standards have some drawbacks. The SOAP messages in transport will be attacked by XML injection attacks and other Web attacks. Therefore, this paper designed a new SOAP message security transport mechanism which added the SOAP Validation node into the existing Web services security transport framework based on the WS standards. At last the experiments demonstrate that this security transport mechanism can truly detect some of XML attacks and improve the security of SOAP message.

Keywords SOA, SOAP, Security transport mechanism, XML injection attacks

1 引言

面向服务的体系结构(SOA, Service-Oriented Architecture)是一个组件模型,它将各个应用程序中不同的功能单元(称为服务)通过这些服务之间定义的良好接口和契约联系起来。接口采用中立的方式定义,它独立于实现服务的硬件平台、操作系统和编程语言,使得构建在各种系统中的服务以一种统一和通用的方式进行交互[1,2]。

SOA使分布式应用集成更加快捷、灵活,同时使其应用系统的安全保障更加困难。SOA虽然解决了异构平台下Web服务的互相调用,但由于需使用XML表示的SOAP(Simple Object Access Protocol,简单对象访问协议)消息进行信息传递与服务调用,因此SOA系统不仅会受到传统的网络攻击(如基于TCP/IP协议),还会受到基于SOAP消息的攻击,如XML注入攻击、强制解析攻击以及各种DOS攻击等[3,4]。传统的防火墙、IDS(Intrusion Detection System,入侵检测系统)等安全手段基于TCP/IP包过滤技术,因而无法检测与预防[5]使用XML表示SOAP消息的SOA系统中的新攻击。

本文以SOA为基础,对现有基于WS安全标准的Web服务传输机制进行深入分析,提出在SOAP消息中添加SOAP Validation节点以改进现有传输机制,然后在.NET平台上设计和实现了这种新的安全传输机制,并证明了它能有效检测出XML注入攻击这种经典的Web攻击。

2 SOAP消息安全传输机制分析

2.1 SOAP消息传输的安全性要求

当SOAP消息在Internet上交互时,存在着信息被监听篡改的风险。为了保证信息传输的安全性,在进行SOAP消息交换时有以下3点基本要求[6,7]:

到稿日期:2011-07-17 返修日期:2011-09-30 本文受国家自然科学基金(60979011),天津市应用基础与前沿技术研究计划项目(09JCYBJC02300),中国民航大学科研启动基金项目(07QDXX)资助。

华 悦(1987-),男,硕士生,主要研究领域为SOA及Web服务安全,E-mail:huayuehuayue@126.com;徐 涛(1962-),男,教授,博士生导师,主要研究领域为软件体系结构/智能信息处理。

完整性：保证信息在传输过程中不被偶然或故意破坏，保持信息的完整、统一。特别是指没有经过授权的用户不能改变信息内容。

机密性：保证没有经过授权的用户实体或进程无法窃取信息。

不可否认性：保证信息的发送者对信息的发送不能抵赖或否认。

2.2 采用 WS 安全标准的 Web 服务安全框架

WS 安全标准定义了一个端对端的安全处理框架。该安全框架采用 XML 签名结合安全令牌，以保证消息在传输过程中不会被更改。同时，安全框架采用 XML 加密结合安全令牌以保证 SOAP 消息的一部分是保密的。最后，WS 安全标准用一组安全头将 XML 签名片段与 XML 加密片段等封装在 SOAP 消息中[8]。

图 1 给出了一个采用 WS 安全标准的 Web 服务安全传输框架。消息的发送者(图 1 中的 Web 服务请求者)先从安全令牌服务机构获取到所需要的安全令牌，根据 SOAP 协议产生满足其策略的 SOAP 消息。其中 SOAP 消息的〈Security〉首部中添加用 WS 安全标准定义的完整性和机密性信息(首部块中允许附加与安全相关的信息)，然后将整个 SOAP 消息发送给消息接收者。接收者既可以是最终的消息接收者(图 1 中的 Web 服务提供者)，也可以是消息的中间转发节点[9]。

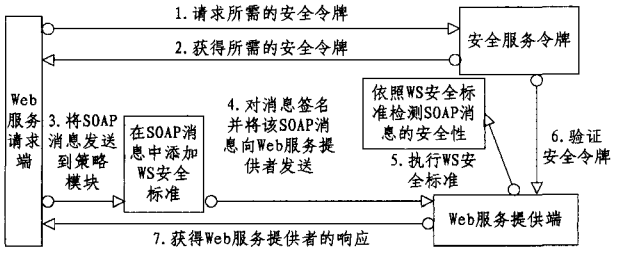

图 1 采用 WS 安全标准的 Web 服务安全传输框架

然而，WS 安全标准的应用建立在服务请求者和服务提供者相互信任的基础上。Web 服务请求者在调用 Web 服务时，请求者和提供者通过用 XML 表示的 SOAP 消息来进行通信。如果在这些 SOAP 消息中插入任何攻击片段，那么传统的基于 TCP/IP 的防火墙无法检测攻击。此外，一个 SOAP 消息在传输过程中可以被数个中间节点接收并处理，然后传给后继节点。而传输层的安全传输机制 SSL/TLS 只能提供点对点的安全传输，不完全适合 SOAP 消息跨节点保护。另外，SOAP 消息的某些特性还会带来很多新的 XML 攻击。

2.3 两种针对 SOAP 消息的攻击

SOAP 消息在传输过程中存在着很多恶意攻击者，它们监视网络中传输的 SOAP 消息。截获 SOAP 消息后，更改 SOAP 消息中的内容和结构，将未验证的数据写入 SOAP 消息。修改过后的 SOAP 消息虽然符合语法要求，但其中原有的数据不能被正确地解析出来。

2.3.1 XML 过载攻击

服务提供者在处理服务请求者发送的消息时，首先要解析 SOAP 消息。而 XML 过载攻击[10]指攻击者在截获 SOAP 消息后对数据大量添加参数，使得 XML 文档变得十分冗长而难以解析，耗尽机器有限的硬件资源。

例：攻击者可以嵌套无穷无尽的标签

```
〈Envelope〉
  〈Body〉
    〈sum〉
      〈a〉3〈/a〉
      〈b〉2〈/b〉
〈b〉2〈/b〉
〈b〉2〈/b〉
  …
```

传统防御此类攻击的办法是限制进入服务提供者的 SOAP 消息的大小，然而这并不能保证进入到服务器端的 SOAP 消息不包含 XML 过载攻击。

2.3.2 XML 注入攻击

XML 注入攻击指攻击者将未验证的数据写入 SOAP 消息。这里给出一种通过更改 XML 嵌套结构的 XML 注入攻击，即在一个元素中嵌套另外一个元素进行攻击[11]。如

```
〈Envelope〉
  〈Body〉
    〈sum〉
      〈a〉〈b〉3〈/b〉〈/a〉
      〈b〉3〈/b〉
```

攻击者截获 SOAP 消息后，把元素〈b〉3〈/b〉当成参数嵌入到元素 a 中间。按照该 Web 服务的服务描述语言 WSDL (Web Services Description Language)规定，元素 a 应该是一个 int 型的数据，而受攻击后元素 a 却嵌套了一个子元素 b，显然这个 SOAP 消息不符合传输过程中对于消息完整性的要求，它应该被抛弃。但是在. NET 平台下，该 SOAP 消息却能被正常解析，实验得出的解析结果为 a=3；b=0。

3 SOAP Validation 消息安全传输机制

3.1 SOAP Validation 节点

SOAP 消息可以在传输过程中被扩展(如安全头等)[12]，但扩展部分并没有包含与 SOAP 消息数据结构相关的信息，而攻击者利用最多的恰恰是 SOAP 消息的数据结构。所以引入一个称为 SOAP Validation 的节点，可以用于保存 SOAP 消息的数据结构信息，并将其添加到 SOAP 消息中一并传输。

对 XML 过载攻击和 XML 注入攻击进行仔细分析，可以得出以下结论：

两种攻击都是对 SOAP 消息中的 XML 元素进行修改(在 SOAP 消息的元素中删除或在后面附加一些内容，或者添加一些元素)。

SOAP 消息中的 XML 元素被攻击后，导致的意外修改会使得原 SOAP 消息中某些元素的前驱后继关系和数量发生改变，这样原有 SOAP 消息中元素的层次结构也发生改变。

SOAP Validation 节点的作用是保存 SOAP 消息中参数的信息结构(如参数元素的个数、名称以及层次信息等)。这些信息可以在 SOAP 消息的发送程序创建该 SOAP 消息的同时进行计算，因此不会引起额外的开销。

3.1.1 SOAP Validation 的结构

用于检测 XML 注入攻击和 XML 过载攻击的 SOAP

Validation 节点内容包括参数子元素的个数(按名称)、参数子元素的名称、参数子元素的前驱后驱以及兄弟关系。

为了获取这些内容,在 SOAP 消息发送者处设置一个过滤器,通过它来获得 SOAP 传输前消息中相关参数的结构和数值,并记录下这些相关信息。

3.1.2 针对 SOAP Validation 节点的攻击

添加了 SOAP Validation 节点的 SOAP 消息,在传输过程中,其节点本身也可能被攻击者篡改,导致无法检测出部分 XML 攻击。针对这种情况,可以通过读取服务提供者处 WSDL 文档中的参数信息,与 SOAP Validation 节点中信息进行对比来发现可能针对 SOAP Validation 节点的攻击。

3.2 SOAP Validation 安全传输机制的设计

图 2 给出添加了 SOAP Validation 节点的 Web 服务安全框架。与图 1 的差别在于,在传输过程中增加了 SOAP Validation 信息添加模块和 SOAP Validation 信息检测模块。当 SOAP 消息传输前,调用 SOAP Validation 信息添加模块来向 SOAP 消息添加 SOAP Validation 节点;当消息到达中间节点和最终接收端时,调用 SOAP Validation 信息检测模块来对 SOAP 消息进行安全检测。

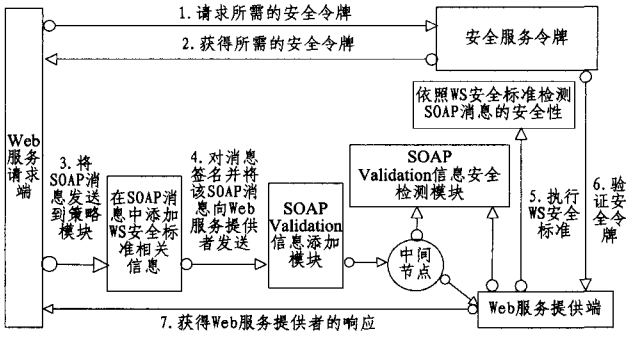

图 2 添加了 SOAP Validation 处理节点后的 Web 服务安全传输框架

3.3 用 SOAP Validation 安全传输机制检测 XML 攻击

为了检测 XML 注入攻击和 XML 过载攻击,图 2 的中间节点接收 SOAP 消息后,SOAP Validation 信息检测模块提取 SOAP Validation 节点中的数据结构信息,与接收到的 SOAP 消息中的参数信息进行对比检测。若发现两者不同,安全检测模块立即输出异常,警告接收到的 SOAP 消息受到攻击。

当 SOAP 消息最终到达服务接受者处时,为了防止针对 SOAP Validation 节点的攻击,SOAP Validation 信息检测模块首先读取 WSDL 文档,获取 WSDL 文档中的参数信息,将其与 SOAP Validation 节点中的信息进行对比,检测 SOAP Validation 节点是否受到攻击。若确认 SOAP Validation 节点未受攻击,则再将 SOAP Validation 节点中的信息与 SOAP 消息中的参数信息进行对比检测,确保服务提供者接收到的 SOAP 消息绝对安全。

4 SOAP Validation 安全传输机制检测 XML 攻击实证

由于 XML 注入攻击远比 XML 过载攻击复杂,在解决 XML 注入攻击的同时 XML 过载攻击也可以迎刃而解,因此本节选取 XML 注入攻击作为实验对象。为了检测更改结构的 XML 注入攻击,首先在 VS2005 平台下建立一个两数相加的 Web 服务 sum[13],随后模拟攻击场景,以获取受攻击的 SOAP 消息,最后利用 SOAP Validation 安全传输机制来检测攻击。

4.1 建立服务

首先建立一个两数(a 和 b)相加的 Web 服务 sum,并截获其产生的 SOAP 消息。图 3 是一段未受到攻击的 SOAP 消息片段,其中元素 a 是一个值为 3 的 int 型。

```
<soap: Body>
  <sum XMLns="http://tempuri.org/">
      <a>3</a>
      <b>2</b>
  </sum>
</soap: Body>
```

图 3 受攻击之前部分 SOAP 信息

4.2 添加 SOAP Validation 节点

在.NET 框架下,可以通过调用 System.xml 库类中的一些函数来获取 SOAP 消息传输前(未受攻击)的参数信息,该库类中另一些函数可以实现向 SOAP 消息中添加 XML 子节点。SOAP Validation 节点就是通过这些函数构造的。图 4 是添加了 SOAP Validation 节点后的 SOAP 消息。SOAP Validation 信息如下:

a)参数个数为 2;

b)第一个元素名称为"a";

c)第一个元素子节点为一个具体值("#text");

d)第一个元素直接后驱为"b";

e)第一个元素直接前驱为"无"。

```
<soap: Envelope>
      <soap: Header>
    .......
  </soap: Header>
  <soap: Body>
    <sum XMLns="http://tempuri.org/">
      <a>3</a>
      <b>2</b>
    </sum>
  </soap: Body>
</soap: Envelope>
   <SOAP Validation ID="请求的元素个数" number="2"
第一个元素名称="a" 第一个元素子节点="#text"
第一个元素直接后驱="b" 第一个元素直接前驱="无" />
```

图 4 添加 SOAP Validation 节点后的 SOAP 消息

4.3 受 XML 注入攻击的 SOAP 消息

图 5 是受攻击后的 SOAP 消息片段(已经添加了 SOAP Validation 节点),这里元素 a 不再是 int 型,而是嵌套了一个子元素 b。如果不经检测直接在 VS2005 平台下进行解析,结果元素 a 值为 2、元素 b 值为 0,这将导致运算结果错误。

```
<soap: Envelope>
 <soap: Header>
    .......
 </soap: Header>
<soap: Body>
   <sum XMLns="http://tempuri.org/">
       <a>
        <b>2</b>
       </a>
       <b>2</b>
   </sum>
 </soap: Body>
</soap: Envelope>
<SOAP Validation ID="请求的元素个数" number="2"
第一个元素名称="a" 第一个元素子节点="#text"
第一个元素直接后驱="b" 第一个元素直接前驱="无" />
```

图 5 添加 SOAP Validation 节点后受攻击的 SOAP 消息

4.4 检测 SOAP 消息中的 SOAP Validation 节点

在 SOAP 消息被中间节点接收后,需要验证 SOAP Vali-

dation 节点中参数信息，安全检测模块读取所收到的 SOAP 消息。参数的数据结构如下：

a)参数个数为 3；

b)第一个元素名称为“*a*”；

c)第一个元素子节点为“*b*”；

d)第一个元素直接后驱为“*b*”；

e)第一个元素直接前驱“无”。

可以看出，SOAP Validation 节点中包含的参数信息与 SOAP 消息中的参数数据结构存在差异，于是发出警告，提示 SOAP 消息已受到攻击。

4.5 检测 SOAP Validation 节点是否被更改

为确保 SOAP Validation 节点中的信息在传输过程中未被修改，当 SOAP 消息达到服务提供者处时，可设置过滤器。先从服务器端的 WSDL 中获取参数数据结构，然后与 SOAP 消息中 SOAP Validation 节点中的信息进行对比。

从 WSDL 文档中获取的参数的数据结构信息如下：

a)参数个数为 2；

b)第一个元素名称为“*a*”；

c)第一个元素子节点类型为 int；

d)第一个元素直接后驱为“*b*”；

e)第一个元素直接前驱“无”。

于是，可以看到 SOAP Validation 节点中的数据未被修改，再对 SOAP 消息中的参数数据结构与 SOAP Validation 节点中的信息进行对比(过程与 4.4 节相同)，从而检测 SOAP 消息是否受到攻击。

结束语 本文针对现有的基于 WS 安全标准的安全传输机制进行深入的探讨，指出其存在的缺陷，并提出了一种在原有安全传输机制基础上添加 SOAP Validation 节点的 SOAP Validation 安全传输机制。SOAP Validation 安全传输机制与基于 WS 安全标准的安全传输机制相比，能有效地检测出 XML 注入攻击和 XML 过载攻击等 Web 攻击，进一步确保了 SOAP 消息传输的安全性。更重要的是，本安全传输机制可以通过增加 SOAP Validation 节点中的内容来检测出更多的 XML 攻击。其次，本安全传输机制在 SOAP 消息经过中间节点时，便可对 SOAP 消息进行安全检测，从而在第一时间检测出 XML 攻击，提高了检测的效率。但是本安全传输机制还无法对其他一些 Web 服务攻击(如签名包装攻击)进行有效检测，需进一步完善，这是本课题下一步的研究工作。

参 考 文 献

[1] Nikolao A, Island C. Service Oriented Architecture: Tools and Technologies[C]//Proceedings of the 11th WSEAS International Conference on Computers. 2007:485-490

[2] Bhargavan K, Fournet C, et al. An Advisor for Web Services Security Policies[C]//Proceedings of ACM Workshop on Secure Web Services(SWS). 2005:1-9

[3] Xu Tao, Yi Chun-xiao. SOAP-based Security Interaction of Web Service in Heterogeneous Platforms [J]. Journal of Information Security, 2010, 2(1):1-7

[4] Xu Tao, Yi Chun-xiao. Signature and Encryption on Parts of SOAP Message Based on Rampart[C]//Proceedings of 2nd International Conference on Intelligent Systems and Applications. 2010:1218-1223

[5] 凌晓东. SOA 综述[J]. 计算机应用与软件，2007，24(10)：122-123

[6] Viega J. Why Applying Standards to Web Services is not Enough[C]//Proceedings of IEEE Security and Privacy. 2006:25-31

[7] 刘振鹏，周冬冬，等. 一个基于 SOAP 消息的 Web 服务综合安全模型[J]. 武汉大学学报：理学版，2006，52(5)：570-573

[8] 李双江，郝克刚，葛伟. 一种基于 Axis2 的 SOAP 安全传输模型的研究[J]. 计算机技术与发展，2008，18(11)：142-145

[9] 程睿，郑有才. 基于 SOA 的 SOAP 消息交互安全机制的研究与实现[D]. 西安：西安电子科技大学，2008

[10] Jensen M, Gruschka N, Hekenhoer R, et al. SOA and Web Services: New Technologies, New Standards-New Attacks[C]//Processings of the 5th IEEE European Conference on Web Service. 2007:11-13

[11] 夏敏，陈平. Web 服务安全关键技术研究综述[J]. 网络安全技术与应用，2009，3(3)：69-71

[12] 付永军. 基于 WSE3.0 的 Web 服务安全的应用研究[D]. 成都：西南交通大学，2007

[13] 孟军，盛雨，刘洪波. 基于. NET 的 SOAP 加密方法研究与实现[J]. 计算机科学，2005，32(8)：52-54

(上接第 53 页)

[4] Li Zhi-jie, Huang Fei-xue. Grid Resource Allocation Based on Evolutionary Algorithm[C]//Fourth International Conference on ICNC '08. 2008:455-459

[5] 黄智兴，刘洪涛，邱玉辉. 基于经济机制的资源管理与网格经济研究进展[J]. 计算机科学，2006，33(9)：110-114

[6] Vanderster D C, Dimopoulos N J, Parra-Hernandez R, et al. Resource allocation on computational grids using a utility model and the knapsack problem[J]. Future Generation Computer Systems, 2009, 25(1):35-50

[7] Huang P, Peng H, Lin P, et al. Macroeconomics based Grid resource allocation [J]. Future Generation Computer Systems, 2008, 24(7):694-700

[8] Li Z J, Cheng C T, Huang F X. Utility-driven solution for optimal resource allocation in computational grid[J]. Computer Languages Systems & Structures, 2009, 35(4):406-421

[9] Bredin J, Maheswaran R T, Imer C, et al. Computational markets to regulate mobile-agent systems[J]. Autonomous Agents and Multi-Agent Systems, 2003, 6(3):235-263

[10] Subrata R, Zomaya A Y, Landfeldt B. A cooperative game framework for QoS guided job allocation schemes in grids[J]. IEEE Transactions on Parallel and Distributed Systems, 2008: 1413-1422

[11] 李志洁，程春田，黄飞雪，等. 一种基于序贯博弈的网格资源分配策略[J]. 软件学报，2006，17(11)：2373-2383

[12] 李志洁，程春田，黄飞雪. 基于合作博弈的网格资源分配[J]. 大连理工大学学报，2007，47(6)：909-913

[13] 涂志勇. 博弈论[M]. 北京：北京大学出版社，2009

[14] Carroll T E, Grosu D. Formation of virtual organization in grids: a game-theoretic approach[J]. Concurrency and Computation-Practice & Experience, 2010, 22(14):1972-1989