# 一种基于 SOAP 扩展的 SOAP 消息分析方法①

李林静[1]，赵建伟[2]

[1](衢州学院 电气与信息工程学院，衢州 324000)
[2](衢州职业技术学院 信息与电力工程系，衢州 324000)

**摘　要**：提出了一种基于 SOAP 扩展的 SOAP 消息分析方法。首先对使用 MSSoapT 的 SOAP 消息分析方法进行了讨论，接着介绍了 SOAP 扩展实现 SOAP 消息分析的机理，然后重点分析了利用 SOAP 扩展实现 SOAP 消息分析的关键技术，并给出在.NET 平台下实现 SOAP 消息分析的实例。实验结果表明，该方法的实现为设计和验证 Web 服务安全策略提供了一种重要的辅助手段。

**关键词**：MSSoapT；SOAP 扩展；SOAP 消息分析；Web 服务安全策略

## Method for SOAP Message Analysis Based on SOAP Extension

LI Lin-Jing[1], ZHAO Jian-Wei[2]

[1](College of Electrical and Information Engineering, Quzhou University, Quzhou 324000, China)
[2](Department of Information and Electric Power Engineering, Quzhou Vocational and Technical College, Quzhou 324000, China)

**Abstract**: This paper presented a method for SOAP message analysis based on SOAP extension. The paper discussed the method of SOAP message analysis using MSSoapT. Firstly, it introduced realization mechanism of SOAP message analysis using SOAP extension, then mainly analyzed key technologies how to realize SOAP message analysis using SOAP extension, gave a instance of SOAP message analysis in .NET. The results show that Implementation of the method for the design and verification of Web services security strategy provides an important adjunct.

**Key words**: MSSoapT; SOAP extension; SOAP message analysis; Web services security strategy

随着 Web 服务在企业应用集成、电子商务、电子政务等领域的广泛应用，其安全问题日益突出，已成为目前信息安全的研究热点。在整个 Web 服务的通信过程中，几乎所有的消息都要通过 SOAP 来传递，SOAP 消息的安全对整个 Web 服务的安全有着至关重要的影响。近年来，对 Web 服务中 SOAP 消息传递的安全性研究受到了广大研究人员的青睐[1-3]。SOAP 消息分析可以有效地验证、测试设计 Web 服务安全策略的效果和可行性，是设计和实现 Web 服务安全策略中的重要任务之一。

目前国内外 SOAP 消息分析的常用方法是使用 MSSoapT（Microsoft SOAP Toolkit）[4-6]来获取和分析进出 Web 服务 SOAP 消息，在用 MSSoapT 获取 SOAP 消息时，该工具工作于客户端或服务端，对进入客户端或服务端的本地端口进行监听，获取进出客户端或服务端的 SOAP 消息。在对带有安全机制的 SOAP 消息进行监听时，如监听加密后 SOAP 请求和 SOAP 响应，MSSoapT 它在客户端只能获取客户端加密后发出的 SOAP 请求消息和服务端返回的 SOAP 响应消息，它无法在单机上获得到达服务端的 SOAP 请求消息或服务端发送的 SOAP 响应，也就无法查看服务端是否对 SOAP 请求消息解密成功，是否加密 SOAP 响应以及加密 SOAP 响应的效果等，这给在单机上调试 Web 服务安全策略带来了不足，所以利用 MSSoapT 获取和查看没有安全机制的 SOAP 消息是可行的，但在调试 Web 服务安全方案时就得使用 SOAP 扩展技术。基于以上分析，本文提出使用 SOAP 扩展技术来对设计的 Web 服务安全策略进行分析，以讨论进出

① 基金项目:衢州市科技计划(20081030);2010 年浙江省高校优秀青年教师资助计划(浙教办高科(2010)175 号)
收稿时间:2011-01-09;收到修改稿时间:2011-02-24

Web 服务的 SOAP 消息是否遵循 WS-Security 规范[7]。实践应用表明，本文利用 SOAP 扩展实现 SOAP 消息分析的方法为设计和验证 Web 服务安全策略提供了一种重要的辅助手段。

## 1 SOAP扩展实现SOAP消息分析的原理

### 1.1 获取 SOAP 消息分析的时机

SOAP 扩展可以插入 ASP.NET 框架的基础结构以在每个序列化和反序列化阶段之前和之后检查或修改 SOAP 消息，允许开发者在这些消息序列化之前和之后执行加密和签名操作。SOAP 扩展的运行依赖于把 SOAP 消息处理分为各个阶段[8]，每个阶段都是 SoapMessageStage 枚举中的一个值，SOAP 消息传递的各个阶段以及发生时序如图 1 所示。

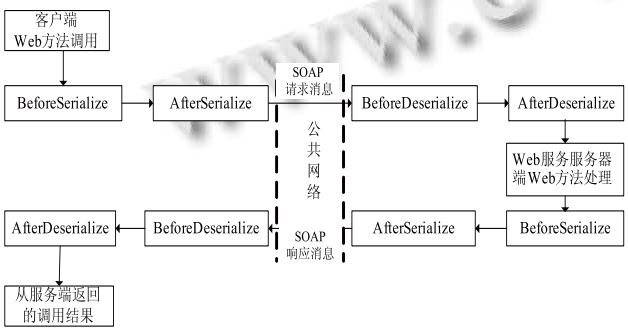


图 1 SOAP 消息传递的各个阶段

在图 1 中，BeforeSerialize 表示恰好在序列化 SoapMessage 之前的阶段，AfterSerialize 表示恰好在序列化 SoapMessage 之后，但在通过网络发送 SOAP 消息之前的阶段，BeforeDeserialize 表示恰好在将 SoapMessage 从通过网络发送的 SOAP 消息反序列化到对象之前的阶段，AfterDeserialize 表示恰好在将 SoapMessage 从 SOAP 消息反序列化到对象之后的阶段。在 BeforeSerialize 和 AfterDeserialize 阶段，SOAP 消息被打包为一个 SoapMessage 对象，而在 AfterSerialize 和 BeforeDeserialize 阶段，SOAP 消息以 XML 形式存在。从这些消息处理阶段可以看出，获取和分析 SOAP 消息的时机在 AfterSerialize、BeforeDeserialize 阶段。

### 1.2 .NET 框架对 SOAP 扩展的支持

在.NET 框架中，SOAP 扩展用 SoapExtension 派生的类表示，实现 SOAP 扩展的全部内容就是定义一个 SoapExtension 类，重载实现它的方法。创建 SOAP 扩展的基本步骤[9]如下：

(1) 从 SoapExtension 派生一个类；

(2) 保存对表示将来 SOAP 消息的 Stream 的引用；

(3) 初始化 SOAP 扩展的数据；

(4) 在相关的 SoapMessageStage 或多个阶段中处理 SOAP 消息；

(5) 配置 SOAP 扩展，使其与特定 Web Services 方法通信。

## 2 运用SOAP扩展实现SOAP消息分析的关键技术

### 2.1 利用 SOAP 扩展建立 SOAP 消息分析的通信平台

在.NET 平台下，利用 SOAP 扩展搭建了分析 SOAP 消息的通信平台，该通信平台包括调用 Web 服务方法的客户端和提供 Web 服务的服务端，客户端对 Web 服务调用的一次完整执行过程如图 2 所示。

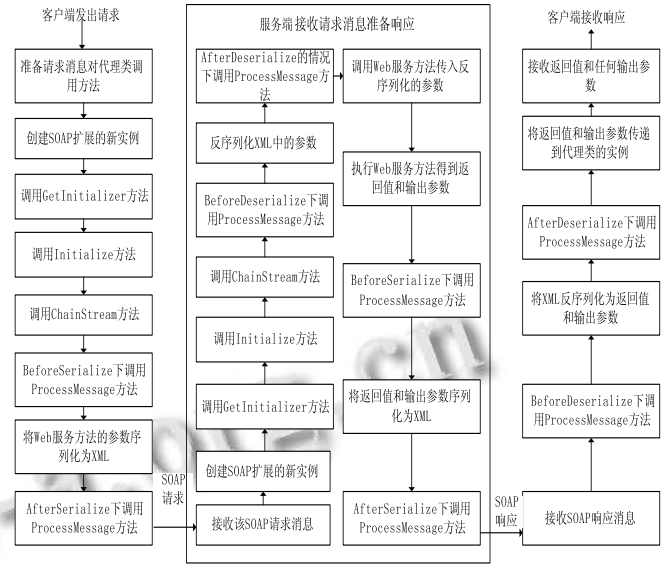


图 2 客户端调用 Web 服务流程

### 2.2 SOAP 消息分析中消息流的操作

在图 2 中，使用 SOAP 扩展可以在 AfterSerialize 和 BeforeDeserialize 阶段直接操作 SOAP 消息，但这之后若再需要访问修改后的 SOAP 请求或响应消息流时，会出现无法读取，因为这个流已经被绑定到 HTTP 传输给了对方，为了解决这个问题必须再创建一个新的内存流对象，这样一个流 oldStream 用于存放 SOAP 请求或响应，另一个流 newStream 成为 SOAP 扩展可以修改的新内存缓冲区，在处理 SOAP 消息方法的各个阶段就对 newStream 进行检查和修改，并把修改后

内容保存到 oldStream 中。所以在 SOAP 扩展的派生类中需定义两个私有成员 oldStream 和 newStream，如下：
private Stream oldstream; private Stream newstream;
重载 ChainStream 方法的具体实现如下，即需要存储对传入流的一个引用同时返回一个新的流引用。

```
public override Stream ChainStream( Stream stream ){
    oldStream = stream;
    newStream = new MemoryStream();
    return newStream;}
```

例如在图 2 中，SOAP 扩展在 BeforeDeserialize 阶段对 newStream 进行了引用，SOAP 扩展对 HTTP 流进行引用的示意图如图 3 所示。在 BeforeDeserialize 阶段之后，必须确保 oldStream 引用的 MemoryStream 对象存放的必要信息，因为它将在 AfterDeserialize 阶段之前被读取。

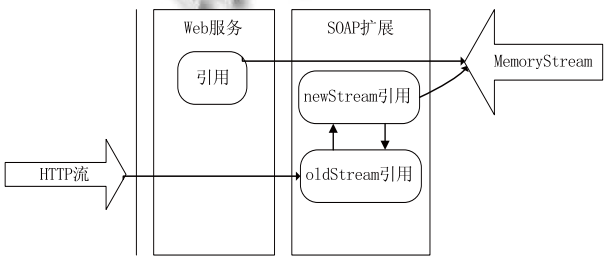


图 3　SOAP 扩展对 HTTP 流引用示意图

### 2.3 重载 ProcessMessage 方法实现带安全机制的 SOAP 消息分析

为了对带安全机制的 SOAP 消息进行分析，需要重载 ProcessMessage 方法，该方法根据 SoapMessage 传递进来的不同阶段对 SOAP 消息进行查看，从上分析可知需在 AfterSerialize、BeforeDeserialize 阶段截获 SOAP 消息。在 ProcessMessage 方法中，可以用如下逻辑代码实现对 SOAP 消息的截获。

```
public override void ProcessMessage(SoapMessage message)
    {switch (message.Stage)
    {case SoapMessageStage.BeforeSerialize: break;
    case SoapMessageStage.AfterSerialize:
    writeoutput(message);break;
    case SoapMessageStage.BeforeDeserialize:
    writeinput(message);break;
    case SoapMessageStage.AfterDeserialize: break;} }
```

自定义方法 writeoutput(message)在流 newStream 中实现对 SOAP 消息进行加密和签名，把修改后结果保存到流 oldStream 中。自定义方法 writeinput(message)把 oldStream 内容复制到 newStream 中，在 newStream 中对指定的部分进行安全性检查如对 SOAP 消息进行解密和签名验证操作。这里涉及到对 newstream 内存缓存区的读取和写入操作，凡是对 newstream 内存缓存区的读取和写入操作之前，都将读取和写入位置 Position 置为 0，即 newstream.Position=0，这一点至关重要，否则无法获取要修改的 SOAP 消息。

## 3　运用Soap扩展实现SOAP消息分析实例

为了验证本文提出的基于 SOAP 扩展实现 SOAP 消息分析方案的可行性，Web 服务安全策略对 SOAP 消息采用 XML 加密，方案选用对称加密算法 AES 和非对称加密算法 RSA 组合策略[10,11]。通信平台中截获 SOAP 请求 Body 中要加密的元素<Purchase Book Order>如下：

```
<PurchaseBookOrder xmlns="http://tempuri.org/">
<orderID>97873021-8</orderID>
<bookname>ASP.NET 程序设计</bookname>
<author>马瑞新</author>
<press>清华大学出版社</press>
<date>2009-05-01T20:14:56.25+08:00</date>
<amount>80</amount></PurchaseBookOrder>
```

在客户端执行触发调用 Web 服务方法的事件后，获取客户端加密后发出的 SOAP 请求消息如下：

```
<EncryptedData
Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
<EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#
s256-cbc"/>-<KeyInfo
xmlns="http://www.w3.org/2000/09/xmldsig#">
-<EncryptedKey
xmlns="http://www.w3.org/2001/04/xmlenc#">
<EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rs
a-1_5"/>-<KeyInfo
xmlns="http://www.w3.org/2000/09/xmldsig#">
```

<KeyName>rsaKey</KeyName></KeyInfo>
-<CipherData><CipherValue>JaeKo3Hv+4V5YTH
sbfz6xbUP...</CipherValue></CipherData>
</EncryptedKey></KeyInfo>-<CipherData>
<CipherValue>Lm8R8UJiNTvemgosGNiDfT4s1Pl
6iysu5h8zwVHVX...</CipherValue>
</CipherData></EncryptedData>

由上可知，加密后的结果遵循 XML 加密规范[12]，与未加密的 SOAP 请求消息相比，<EncryptedData>元素替换了<PurchaseBookOrder>，加密产生的数据放在了<EncryptedData>元素中，<EncryptionMethod>元素指定了加密时使用的算法，<CipherData>元素中的内容为加密<PurchaseBookOrder>元素及子元素的结果，<KeyInfo>元素描述了密钥名称、密钥值等。

服务端在 BeforeDeserialize 之后获取传递过来的 SOAP 请求消息查看其内容与通信平台中截获 SOAP 请求 Body 中要加密的元素相同，这说明服务端接收 SOAP 请求消息后成功实现了 SOAP 请求消息解密。

从上面的消息分析实例可以看出，为了验证设计的 Web 服务安全策略的效果，借助了.NET 搭建的 SOAP 消息分析平台。依次在 AfterSerialize 之前和之后分别截获 SOAP 请求 Body 中要加密的元素、SOAP 请求 Body 中已加密的结果，在 BeforeDeserialize 之后获取传递过来的 SOAP 请求消息，从截获的数据发现在客户端已成功完成对指定内容的加密，在服务接收端对已接收到的 SOAP 消息已成功完成了解密。若采用 MSSoapT 来验证上面设计的 Web 服务安全策略，假定在客户端监听，它在单机上只能获取客户端在 AfterSerialize 之后 SOAP 请求 Body 中已加密的结果，无法获取在 AfterSerialize 之前 SOAP 请求 Body 中要加密的元素，也无法获取到达服务接收端后传递过来的 SOAP 请求消息，也就无法判断达到的 SOAP 请求是否解密成功，这给单机上调试设计的 Web 服务安全策略带来了不足。

## 4 结语

运用 SOAP 扩展在.NET 平台下实现了进出 Web 服务 SOAP 消息安全分析方法。实验结果表明，该方法为研究 Web 服务安全策略是否可行提供了一种验证和测试手段，为 SOAP 消息安全分析提供了一个新的思路和方法。

## 参考文献


1 刘志都,贾松浩,詹仕华.SOAP 协议安全性的研究与应用.计算机工程,2008,34(5):142−144.

2 雷晟.Web Services SOAP 消息安全研究.北京:北京化工大学,2009.

3 Gruschka N, Jensen M, Dziuk T. Event-based Application of WS-Security Policy on SOAP Message. Nov. 2007 Proc. of the 2007 ACM workshop on Secure Web Services.

4 祝伟华,周颖,杨丹.Web 服务的安全性研究.计算机机科学,2005,32(6):76−78.

5 Boncella RJ.Web Services and Web Services Security. Communications of the Association for Information Systems, 2004:344−363.

6 使用 MSSOAPT 查看.NET Web Service 的 SOAP 信息. http://blog.csdn.net/shuaiwang/archive/2007/04/14/1564385.aspx,2007.04.

7 WS-Security v1.1. OASIS. http://www.oasis-open.Org/committees/documents.php?wg_abbrev=wss.

8 陈天煌,李帆.利用 SOAP 扩展实现 Web 服务中 SOAP 消息的安全.武汉理工大学学报,2007,31(3):518−524.

9 刘晓华..NET Web 服务开发指南.北京:电子工业出版社,2002.123−146.

10 顾韵华,傅德胜,王兴.XML 安全技术分析与应用.计算机科学,2009,36(5):118−141.

11 魏海新,张超英.一种灵活的 SOAP 签名加密方案.计算机工程,2010,36(13):151−153.

12 Eastlake D, Reagle J. XML Encryption Syntax and Processing W3C Recommendation. http://www.w3.org/TR/xmlenc-core,2002,12.