

定制 SOAP 协议实现用户身份验证

黄贵平 (中科院研究生院 100080)

摘要:本文通过一个具体的示例,说明了在利用 WEB SERVICE 组件技术构造分布式数据库应用环境下,如何通过定制 SOAP 协议实现用户身份验证。

关键词:Web Service SOAP HTTP XML 数据库访问

1 概述

WEB 服务是一种较新的分布式组件技术,和传统的 WEB 技术要解决人与 WEB 应用程序之间的交互相比,WEB 服务着重解决如何让计算机应用程序来使用 WEB 应用程序所提供的服务,即 WEB 服务。WEB 服务能够提供跨平台、跨语言的服务,它建立在一些列相关 WEB 协议基础之上,它通过标准的 XML 数据编码和通用的 WEB 协议,如 HTTP, SOAP 等,来实现客户端应用程序和 WEB 服务提供者之间的交互。在 WEB 服务技术中最重要一个协议就是 SOAP (Simple Object Access Protocol, 简单对象访问协议),利用 SOAP, 消息可以在 WEB 服务使用者(客户端)和 WEB 服务提供者之间进行传递,从而实现分布式应用环境。笔者在利用 WEB 服务技术开发的一个分布式应用系统中,为了实现 WEB 服务提供者对服务使用者的身份验证,采用了定制 SOAP 协议技术,解决了在特定的 WEB 服务应用环境下,对用户合法性的确认。

2 SOAP 协议分析及定制方法的实现

SOAP 协议是一个基于 XML 在分布式环境中用于交换消息的协议,简单来说,SOAP = HTTP + XML。从协议分层的角度来看 SOAP 和 HTTP 协议一样属于应用层协议,但 SOAP 协议要在 HTTP 协议之上,SOAP 消息要通过 HTTP 协议(也可通过其他协议,如 FTP, SMTP 等)封装后进行传输。SOAP、HTTP、XML 三者之间的关系如图 1 所示。

具体来说,当用户使用 SOAP 协议调用远程 WEB 服务提供的服务方法(即函数)时,客户端应用程序首先和远程 WEB 服务建立 HTTP 连接,将要申请使用的

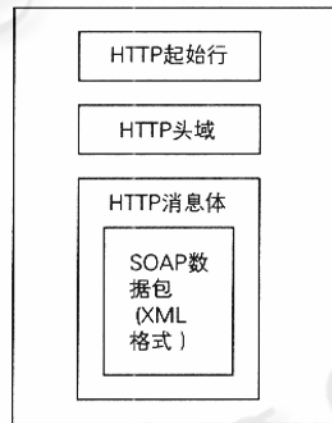


图 1 SOAP 协议、HTTP 协议及 XML 三者之间的关系示意图

WEB 服务的函数名及参数,经 XML 编码后,构成 SOAP 消息,该消息经应用层协议 HTTP 封装(见图 2),再经过传输层协议 TCP/IP 封装后,在 INTERNET 网上传输。客户端 WEB 服务的调用程序在接到远程 WEB 服务返回的 SOAP 消息后,解析其中的 XML 的内容,还原成请求程序可以接受的返回结果。

```
POST /WebService5/Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://NPFPC.GOV.CN/VevifyUser"
<? xml version="1.0" encoding="utf-8"? >
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Header >
```

```

    < MyHeader xmlns = " http://NPFPC. GOV.
CN" >
        < Name > string </Name >
        < Password > string </Password >
    </MyHeader >
</soap:Header >
<soap:Body >
    < VevifyUser xmlns = " http://NPFPC. GOV.
CN" / >
    </soap:Body >
</soap:Envelope >

```

图 2 经过 HTTP 封装后的 SOAP 消息包

说到 SOAP 消息的结构,它首先是 XML 文件,包含构成 XML 文件所要求的相关元素,这些元素在一定的程度上可以看作是对象,每个对象完成不同的作用,SOAP 消息主要包括下列三个元素,即:

- SOAP 消息信封 (Envelope):它是整个 SOAP 消息的根元素,也是 SOAP 必须的,其他两个元素包含在该元素内部。

- SOAP 消息头 (Header):它是 SOAP 消息中的可选元素,程序员可以利用该部分实现诸如用户身份验证、加密、事务处理等功能。

- SOAP 消息体 (Body):是每个 SOAP 消息必须有的部分,SOAP 体包括请求或响应消息的主要内容,包括调用远程 WEB 服务的方法的名称及参数,或方法调用后返回的结果。SOAP 消息结构如图 2 所示。



图 3 SOAP 消息的结构示意图

SOAP 协议规定 SOAP 消息的内容使用 XML 格式,但并没有严格地定义 Header 的内容。在这个 XML

格式化的 SOAP 消息中,有一个名为 <Envelope > 的根标记,这个根标记下面有一个可选的 <Header > 标记和一个必须的 <Body > 标记。对 <Body > 标记我们不能做什么,但是对于可选的 <Header >,我们可以通过程序来在其中适当增加若干个子节点,每个子节点在 .NET 中被称为 SOAP Header。程序可以利用增加的子节点,即 SOAP Header 来传输额外的信息。在下面的示例中,WEB 服务中相关服务函数通过读取客户定制的 SOAP Header 中的授权信息,来决定是否提供相应的服务,对于 WEB 服务中提供的其他的不需要验证授权的服务函数则可以直接忽略 Header 中的授权信息。

3 用户身份验证的具体实现

下面是实现用户身份验证的服务器端 C# 代码及说明:

```

using System;
using System. Web. Services;
using System. Web. Services. Protocols;
using System. Data;
using System. Data. OleDb;
namespace Mynamespace
{
    //定制一个 SOAP Head 类
    public class MyHeader : SoapHeader
    {
        public string Name;
        public string Password;
    }
    [WebService( Namespace = " http://NPFPC. GOV.
CN" )]
    public class MyWebService : System. Web. Serv-
ices. WebService
    {
        //申明一个定制 SOAP Head 类的对象
        public MyHeader loginHead;
        [WebMethod]
        //通过 WebMethod 属性将定制的 SOAP
        Head 应用于 Web Service 的服务
        [SoapHeader ( " loginHead" , Direction =
SoapHeaderDirection. InOut) ]
        //定义一个用户身份验证函数

```

```

public bool VevifyUser( )
{
    if( loginHead == null)
        loginHead = new MyHeader ( );
    if ( Checkusers ( loginHead. Name, login-
Head. Password) )
        return true;
    return false;
}

//定义一个 Web 服务,并将 SOAP Head 中
用户信息与服务器中保存的用户数据库进行比
较,如一致,则可以调用服务
[ WebMethod ] [ SoapHeader ( " login-
Head", Direction = SoapHeaderDirection. InOut) ]
public long Add( int a, int b)
{
    if( VevifyUser( ) )
        return a + b;
    else return 0000;
}

public bool Checkusers ( string name,
string password)
{
    OleDbDataReader dr;
    string MyConnString = " Provider = Mi-
crosoft. Jet. OLEDB. 4. 0; Data Source = e: \hgp \us-
ers. mdb";
    string strSql = " select * from users
where name = " + " " + name + " " + " and pass-
word = " + " " + password + " " ;
    OleDbConnection MyConn = new OleDb-
Connection ( MyConnString );
    OleDbCommand MyComm = new OleDb-
Command ( strSql, MyConn );
    MyComm. Connection . Open ( ) ;
    dr = MyComm. ExecuteReader ( ) ;
    if( dr. HasRows == true)
        return true;
    else return false;
}

```

```

}
}

```

为了通过编程实现定制 SOAP, 需要利用 .NET 类库中的抽象类 SoapHeader 来定义定制的 SOAP Header。程序首先定制一个 SOAP Header, 即类 MyHeader, 此类从 .NET 类库的抽象类 SoapHeader 继承而来, 然后在 MyHeader 中定义公共成员, 在本例中为 Name, Password。经过这样的定义后, 产生的 SOAP Header 如下所示:

```

< soap:Header >
    < MyHeader xmlns = " http://NPFPC. GOV. CN"
    >
        < Name > string </Name >
        < Password > string </Password >
    </MyHeader >
</soap:Header >

```

然后在 WEB SERVICE 类中为使用该定制 Header 的服务方法使用 WebMethod 的 SoapHeader 属性, 如上例中的服务方法 VevifyUser() 和 Add()。程序的工作原理是这样: 当 WEB SERVICE 接收到客户对服务方法 ADD() 的请求时, 先调用 WEB SERVICE 中的 VevifyUser() 方法对用户的合法性进行验证, VevifyUser() 方法读取客户定制的请求 SOAP 消息中的头部, 从中解析出 Name 和 Password, 然后和系统中保存的合法用户进行比较, 如果用户名和密码都正确, 则继续完成服务方法的调用, 否则返回错误代码。

以下是实现用户身份验证的客户端应用程序 C# 代码:

```

using System;
using System. Collections;
using System. ComponentModel;
using System. Data;
using System. Drawing;
using System. Web;
using System. Web. SessionState;
using System. Web. UI;
using System. Web. UI. WebControls;
using System. Web. UI. HtmlControls;
namespace WebApplication19
{

```

```

public class WebForm1 : System.Web.UI.Page
{
    protected System.Web.UI.WebControls.TextBox
    TextBox1;
    protected System.Web.UI.WebControls.TextBox
    TextBox2;
    protected System.Web.UI.WebControls.Label
    ReturnValue;
    protected System.Web.UI.WebControls.Button
    Button1;
    override protected void OnInit(EventArgs e)
    {
        InitializeComponent();
        base.OnInit(e);
    }
    private void InitializeComponent()
    {
        this.Button1.Click += new System.EventHandler(
        this.Button1_Click);
        this.Load += new System.EventHandler(
        this.Page_Load);
    }
    private void Button1_Click(object sender, System.
    EventArgs e)
    {
        localhost.MyHeader header = new local-
        host.MyHeader();
        header.Name = TextBox1.Text;
        header.Password = TextBox2.Text;
        localhost.MyWebService proxy = new local-
        host.MyWebService();
        proxy.MyHeaderValue = header;
        long results = proxy.Add(1,100);
        ReturnValue.Text = results.ToString();
    }
}

```

为了在客户程序的 WEB FORM 中调用 WEB 服务的
服务方法,程序首先必须生成一个 WEB 服务的代理
类,客户端程序对 WEB 服务类中服务方法的访问实际

上是通过这个代理类进行的。代理的生成有两种方法
一种是通过执行命令行程序 WDSL, CSC 产生,另一种
在 Visual Studio .NET 环境中通过引用自动生成,本例
采用后一种办法。在生成的代理类中包含如下定义:

```

public class MyHeader : SoapHeader
{
    public string Name;
    public string Password;
}

```

同时,在代理类中还包括这样的一个声明:

```

public MyHeader MyHeaderValue;

```

客户端的应用程序可以通过代理类生成定制的
SOAP Header 类的对象 header,并对其中的成员变量
Name, Password 赋值,这样,在客户端生成的 SOAP 消
息中就会包括定制的头,其中包括应用程序所要传递
用户信息,如下所示:

```

<soap:Header>
  <MyHeader xmlns="http://NPFPC.GOV.CN"
  >
    <Name>HUANG PING </Name>
    <Password>88823 </Password>
  </MyHeader>
</soap:Header>

```

4 小结

本示例采用了定制 SOAP 协议消息的方法,实现
了在分布式数据库系统应用中,通过 SOAP 协议调用
远程 WEB 服务所提供的服务方法时对调用者身份的
验证,通过对本示例进行适当的修改即可应用于其他
方面,如对数据的加密、事务处理等。

参考文献

- 1 | 王超、张鹏,ASP.NET/XML 深入编程技术,希望电
子出版社,2002. 3.
- 2 | 邹建峰等,C#企业级开发案例精解,人民邮电出版
社,2006. 1.
- 3 | 苏贵洋,ASP.NET 网络编程从基础到实践,电子工业
出版社,2005. 10.
- 4 | 郑淑芬、许嘉仁、张书源,新一代 C#与 ASP.NET 权威
指南,中国青年出版社,2004. 8.