Allen Conway's .NET Weblog

Exploring All Things .NET And Beyond...

Thursday, MAY 24,

Creating a WCF RESTful Service And Secure It Using HTTPS Over SSL

Well I have had a few posts now on security, and focused some specifically on HTTPS and WCF. One of the most popular types of services we can create using WCF are RESTful services. The REST architecture and RESTful services have become so popular due to their inherent flexibility and being more consumer and platform agnostic as opposed to their WCF SOAP WS-* counterparts. However with this popularity and quickly expanding use we must not forget about securing our RESTful services especially when exposing them for wide use on the *internet*. Since REST services communicate over HTTP, we can leverage our existing knowledge and security principals we use for traditional websites that communicate over HTTP.

I wanted to highlight a few options for securing and authenticating to RESTful services using WCF in some upcoming posts. This 1st one will speak to securing the pipeline that REST services communicate over which is HTTP.

As we all know (or if not, now you do!) you can secure the communication over HTTP by using a SSL certificate. This is the 1st step in securing the data being communicated from the client and our RESTful service. Let's begin by looking at our simple ServiceContract. It contains a GET method defined with a CLR name of 'GetCustomerData' being exposed as the noun 'Customer' to adhere to REST architecture standards:

?
```
namespace RESTfulSecuritySH
{
   [ServiceContract]
   public interface ISecureRESTSvcTest
   {

     [OperationContract]
     [WebInvoke(Method = "GET",
                ResponseFormat = WebMessageFormat.Json,
                UriTemplate = "/Customer/{CustID}")]
     Customer GetCustomerData(string CustID);
   }
}
```

Behind the scenes the implemented code is returning the ID provided along a 'Customer' data contract with a some fake customer data. Obviously in a real implementation we would be connecting to a data repository to get this information, but for this example we will just hardcode values to be returned as security is the focus here and not the service implementation details.

?
```
namespace RESTfulSecuritySH
{
  public class SecureRESTSvcTest : ISecureRESTSvcTest
  {
     public Customer GetCustomerData(string CustID)
     {

        Customer customer = new Customer()
        {
          ID = CustID,
```

About Me

Allen Conway

I am a Magenic Senior Consultant that is an advocate of Microsoft .NET technologies and the Microsoft platform both professionally and personally. I enjoy the challenge and creativity behind software engineering, and hope during this process to extract some of my thoughts and ideas in order to give back to others in the community here on this Blog.

VIEW MY COMPLETE PROFILE

Awards

Microsoft Community Contributor 2011 | AWARD

Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Disclaimer

This weblog does not represent the thoughts, intentions, plans or strategies of my employer. It is solely my opinion.

LIVE! 360

Visual Studio LIVE! REDMOND
SET YOUR COURSE FOR 127.0.0.1!
REGISTER NOW
August 18 – 22 | Microsoft Campus | Redmond, WA

Twitter

Follow Me On Twitter

RSS Link

Click here for the RSS link

Tags

.asmx
.ASMX Services
.NET

Contact Me

Blog Archive

- ▶ 2014 (5)
- ▶ 2013 (26)
- ▼ 2012 (26)
  - ▶ December (6)
  - ▶ September (1)
  - ▶ August (1)
  - ▶ July (2)

```
            FirstName = "John",
            LastName = "Smithers",
            PhoneNumber = "800-555-5555",
            DOB = "01/01/70",
            Email = "jsmithers@fancydomain.com",
            AccountNumber = "1234567890"
        };

        return customer;
    }
}

[DataContract]
public class Customer
{
    [DataMember]
    public string FirstName { get; set; }
    [DataMember]
    public string LastName { get; set; }
    [DataMember]
    public string PhoneNumber { get; set; }
    [DataMember]
    public string DOB { get; set; }
    [DataMember]
    public string Email { get; set; }
    [DataMember]
    public string ID { get; set; }
    [DataMember]
    public string AccountNumber { get; set; }
}
}
```

The default configuration for this service uses the **webHttpBinding** as required for RESTful services created using WCF. This configuration is for a self-hosted service as it provides a *baseAddress* value. This identical configuration could be used for an IIS hosted service. Just keep in mind the *baseAddress* will be ignored and replaced by the virtual directory structure set up in IIS.

?

```
<system.serviceModel>
  <services>
           <service  behaviorConfiguration="SecureRESTSvcTestBehavior"
name="RESTfulSecuritySH.SecureRESTSvcTest">

      <host>
        <baseAddresses>
          <add baseAddress="http://DevMachine1234:8099/MyRESTServices/"/>
        </baseAddresses>
      </host>

        <!--webHttpBinding allows exposing service methods in a RESTful
manner-->
      <endpoint address=""
        binding="webHttpBinding"
        behaviorConfiguration="webHttpBehavior"
        contract="RESTfulSecuritySH.ISecureRESTSvcTest" />

      <endpoint address="mex"
        binding="mexHttpBinding"
        contract="IMetadataExchange" />

    </service>
  </services>
  <behaviors>
    <serviceBehaviors>
```

101 Topics
ADO.NET
Advice
Agile
AJAX
Android
Archiecture
Architecture
ASP.NET
asynchronous
Asynchronous Programming
Awards
Azure
Basic Authentication
Best Practices
Bindings
BitLocker
Book Review
Books
Building
Business Logic
C#
Caching
Career
Cassini
Code Camp
code refactoring
Computer Hardware
Conference
Conversions
Custom Built PC
DDD
Debate
Debugging
Decompiling
Delete Files
Design
Design Patterns
DI
DropDownList
Editorial
Encryption
Entity Framework
Enums
Errors
Exceptions
Expression Blend
Expression Encoder
Freebies
FTP
Future Predictions
History
HTML
HTML5
IIS
IIS 7.0
IIS 7.5
Internet Explorer

- ▶ June (1)
- ▼ May (3)
  - Creating a WCF RESTful Service And Secure It Using...
  - Fixing The SignatureMismatch Exception When Updatin...
  - Adding The HTML Target Schema Selection to VS.NET ...
- ▶ April (2)
- ▶ March (6)
- ▶ February (2)
- ▶ January (2)

▶ 2011 (31)
▶ 2010 (42)
▶ 2009 (13)

```
        <behavior name="SecureRESTSvcTestBehavior">
          <serviceMetadata httpGetEnabled="true"/>
          <serviceDebug includeExceptionDetailInFaults="false"/>
        </behavior>
      </serviceBehaviors>

      <!--Required default endpoint behavior when using webHttpBinding-->
      <endpointBehaviors>
        <behavior name="webHttpBehavior">
          <webHttp/>
        </behavior>
      </endpointBehaviors>

    </behaviors>

  </system.serviceModel>
```

At this point if we start our service using the WCF Test Client (if trying this locally), and then call the RESTful URI http://DevMachine1234:8099/MyRESTServices/Customer/8 to our service, you will get the JSON output (below) as expected (use Chrome, FF, or Safari to test. IE and Opera make you open a separate file). 'DevMachine1234' is the name of my dev box and eventually will be the name that needs to match up with the one provided to the SSL certificate which we will discuss in a moment.

{"AccountNumber":"1234567890","DOB":"01\/01\/70","Email":"jsmithers@fancydomain.com","FirstName":"John","ID":"8","LastName":"Smithers","PhoneNumber":"800-555-5555"}

OK, simple enough but notice there *is* data here that really is sensitive. The structure and design of our Customer data contract is not the focus of this post and is a flattened view of example customer data, but regardless some of it is sensitive. Let's now secure the transmission of this data over HTTPS using a SSL certificate.

For this example I am going to use a self-signed certificate I created locally and assign it on my machine to port 8099. I have (2) posts already that explain how to do this: Create A Self-Signed SSL Certificate Using IIS 7 and Applying and Using a SSL Certificate With A Self-Hosted WCF Service.

Let's focus on the configuration changes that had to be made. Now depending on how you decide to host your WCF service there may be subtle differences in the configuration. Also I should mention here that all of the configuration can be done in code if you chose to do so. I typically prefer configuration 1st, and only rely on using code to configure and run my service if there are elements that could be different at runtime and the configuration needs the flexibility of being dynamic. If this is not the case, I lean toward configuration because of its on-the-fly configurability and ease of reading. This is one case where either method works fine.

If using IIS to host your service, the certificate can be added through the IIS Management console and applied to the site instead of using the command line netsh.exe tool for self-hosted services. The 1st configuration change is to update the base address if you are using a self-hosted service to HTTPS:

?
```
<add baseAddress="https://DevMachine1234:8099/MyRESTServices/"/>
```

Next we must add a **bindingConfiguration** value to our endpoint as shown below:

?
```
<endpoint address=""
     binding="webHttpBinding"
     bindingConfiguration="webHttpTransportSecurity"
     behaviorConfiguration="webHttpBehavior"
     contract="RESTfulSecuritySH.ISecureRESTSvcTest" />
```

We must also add in this new binding configuration section. If you add a **bindings** section at the root of the **system.serviceModel** section, we can configure the security to be "Transport". When using Transport security with WCF, it automatically expects we want to use HTTPS for our endpoint. There are (3) different modes in the enumeration for the

introduction
JavaScript
jQuery
Lambda Expressions
Learning
LightSwitch
LINQ
LINQPad
ListBox
Metro Apps
Microsoft
Moq
MS
MSDN
MSEvents
MVC
Natural Language Processing
Non .NET Related
NSIS
NuGet
Object Data Source
OData
ONETUG
OOP
Opinion Piece
Performance Counter
Periodic Execution
PLINQ
Project Methodology
Promotions
Prototyping
Recursion
reflection
REST
Security
Sentiment Analysis
Setup Packages
Silverlight
Smart Phones
Software Engineering
SQL
SQL2000
SQL2008
SSL
System.IO
TechEd
thoughts
Tools
TPL
Training
TSQL
Twitter API
UI Design
Unit Testing
ValueInjecter
VB.NET
VB6
Visual Studio Live

security element: **None**, **Transport**, and **TransportCredentialOnly**. 'None' indicates there is no required security on the endpoint, 'Transport' indicates we want to secure our endpoint with a SSL certificate using HTTPS, and 'TransportCredentialOnly' provides HTTP-based client authentication only and does not provide message integrity and confidentiality. The new endpointConfiguration is below:

?
```
<bindings>
  <webHttpBinding>
    <binding name="webHttpTransportSecurity">
      <security mode="Transport" />
    </binding>
  </webHttpBinding>
</bindings>
```

Next within the service behavior we must *enable* HTTPS and *disable* HTTP. This way our RESTful service can *only* be accessed via HTTPS. This is a very important step if you intend to have your service only accessed in a secure manner. Otherwise you leave the door open to still access it via HTTP. This may be an OK scenario if hosting non-secure data that you want to be able to be accessed via HTTP or HTTPS, but otherwise it should only be one or the other.

?
```
<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>
```

Lastly, we need to update the metadata publishing endpoint to use HTTPS as well:

?
```
<endpoint address="mex"
  binding="mexHttpsBinding"
  contract="IMetadataExchange" />
```

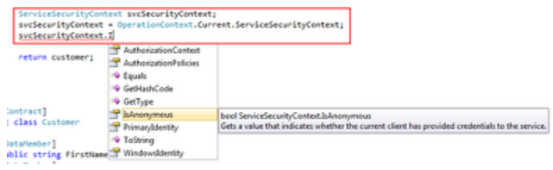
At this point restart your service. If you encounter any errors, it is probably because the SSL certificate does not match the name of your service, the SSL cert has not been applied correctly to the port (self-hosted services only), or the all the configuration updates were not made correctly. If you completed everything successfully, you can now access your site at the following URL: https://DevMachine1234:8099/MyRESTServices/Customer/8

Now try opening a new browser and access it via HTTP; it will not work and this is what we wanted. We have now secured our RESTful service's communication.

One last big ticket item here that will set me up for my next posts on securing RESTful services has to do with the security context before and after securing our service. This will ultimately allow us to *authenticate* our users and move forward with security. Either debugging the RESTful call to your existing 'GetCustomer' method or from a new call that simply returns a string, try looking at the following lines of code:

?
```
//Look at the ServiceSecurityContext both using secured and non-secured
service configurations:
ServiceSecurityContext svcSecurityContext;
svcSecurityContext = OperationContext.Current.ServiceSecurityContext;
```

Notice how if you run your service without Transport security using HTTPS, the ServiceSecurityContext instance will be 'null'. However once we secure our service, this instance will have value. Specifically if you look at the **.IsAnonymous** property, you will notice it is now = **"true"** upon securing our service.


```
ServiceSecurityContext svcSecurityContext;
svcSecurityContext = OperationContext.Current.ServiceSecurityContext;
svcSecurityContext.I

return customer;

ontract]
: class Customer

DataMember]
ublic string FirstName
```

bool ServiceSecurityContext.IsAnonymous
Gets a value that indicates whether the current client has provided credentials to the service.

We will work on populating this value in my next post and moving forward to authenticating the client.

Visual Studio Online
VS.NET 2008
VS.NET 2010
VS.NET 2012
VS.NET 2013
VS.NET IDE
VSLive
VSLive 2013
VSS
WCF
WCF Data Services
Web API
WebForms
Windows 7
Windows 8
Windows Authentication
Windows Server
Windows Service
WiX
WMI
XML

Posted By Allen Conway On 5/24/2012 11:58:00 PM
Labels: REST, Security, SSL, WCF

Related Postings:

- How To: Set Up BitLocker Full Disk Encryption + Pre-Boot Pin In Windows 7 Ultimate
- Applying And Using A SSL Certificate With A Self-Hosted WCF Service
- Tip On Preventing Web Discover Service Error
- Download FTP Files Using FTP Over SSL (SFTP) In .NET
- Visual Studio LIVE! Day 1 - WCF And Web API
- Exposing Multiple Binding Types For The Same Service Class In WCF
- How To Serialize A SyndicationFeed Object To Be Returned From WCF
- How To Fix When A WCF Service Does Not Debug Via The "WCF Test Client" Tool
- Fixing The "Attempting To Access A Service In A Cross-domain Way..." Error When Consuming A WCF Service Running Via A VS.NET 'Localhost' Binding
- Using Postman For Google Chrome To Call REST And OData Services
- Using Basic Authentication In REST Based Services Hosted In IIS
- RESTful Services: Authenticating Clients Using Basic Authentication
- Creating A WCF RESTful Service And Secure It Using HTTPS Over SSL
- Using WCF Data Services Tutorial - Part 1
- Create A Self-Signed SSL Certificate Using IIS 7

10 Comments:

venkat merugu said...

Nice and Needed Post Waiting For Next Post

August 7, 2012 At 8:44 AM

Buy personal lubes said...

Thanks for shared this information and I need this post.

September 12, 2012 At 10:33 AM

vimal patel's blogspot said...

Hi Allen,

I have made the service which is http based and works fine. I want to make it SSL-HTTPS enabled.

I made all the above changes to make the service HTTPS-SSL enabled.

But I start receiving :- Could not find a base address that matches scheme https for the endpoint with binding WebHttpBinding. Registered base address schemes are [http].

The moment I remove, bindingConfiguration="webHttpTransportSecurity" from service endpoint. the above error goes away and gives new one:-

Could not find a base address that matches scheme https for the endpoint with binding MetadataExchangeHttpsBinding. Registered base address schemes are [http].

This error goes away if I remove the "s" from mexhttpsbinding.

which then gives:-
The HttpsGetEnabled property of ServiceMetadataBehavior is set to true and the HttpsGetUrl property is a relative address, but there is no https base address. Either supply an https base address or set HttpsGetUrl to an absolute address.

it goes away if I remove httpsGetEnabled="true"

Thus, after removing all changes, I am not able to achieve the HTTPS-SSL enabled service.

It keeps on saying that registered base address scheme is http and not https.

Any idea what must be the reason. However, I was successfully able to apply the certificate.

Please see the config settings:-

October 24, 2012 At 9:29 AM

wiwi said...

Really help me much, thanks.

December 18, 2012 At 5:06 AM

Kiran said...

Thanks for the nice post sir. It really saved my time a lot.

January 8, 2013 At 1:43 PM

Anonymous said...

Great blog man, thank you so much.

June 23, 2013 At 7:42 PM

Travich said...

Great post, followed it and it was tremendously helpful. Would suggest if you are not familiar with certificate creation to post a link to that. This was my first time to do so and I had to do some research...

August 6, 2013 At 2:33 PM

Anonymous said...

You saved me so many hours of work. I was getting frustrated on how to achieve this. Thanks a ton.

October 8, 2013 At 10:15 AM

Anonymous said...

Thank you. A developers butt has been saved.

December 10, 2013 At 4:59 AM

Allen Conway said...

Great! Saving butts is a good thing :D Glad it helped and thanks for the feedback.

December 10, 2013 At 12:14 PM

Post a Comment

Enter your comment...

Comment as: Select profile...

Publish Preview

Newer Post Home Older Post

Allen Conway's .NET Weblog

StatCounter sitemeter