A complete guide to create secure WCF REST API with custom Basic Authentication

WCF REST API services are still being used by many developers for client server connectivity for data and messaging. This blog is a complete guide on creating a WCF Rest service from scratch and Adding security to the service using Basic Authentication. Then we'll learn how to encrypt the basic authentication information which would be sent over the network using SSL. The main sections of this guide are:

- Creating a WCF REST API service
- Hosting a WCF REST service in IIS from Visual studio (on local machine)
- Deploying a WCF REST service on IIS (Local machine)
- Adding security to the Service by using Basic Authentication
- Securing basic authentication credentials using SSL over Http i.e. (Https)
    - Creating a certificate and Enabling IIS website to use Https
    - Setting up WCF REST service to use SSL (Https)

Pre-requisite:

- Basic knowledge of Visual studio/WCF basics.
- Visual studio version > 2008
- .Net framework > 3.5 installed
- IIS server >7 or equal

Creating a WCF REST API service

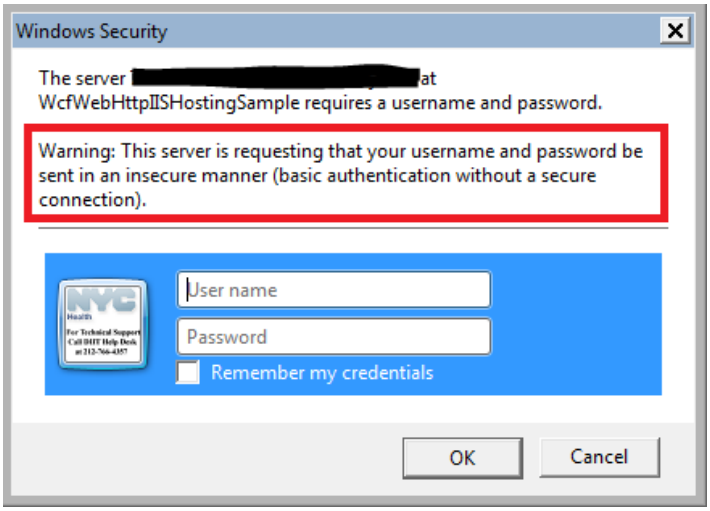
WCF REST service hosting in IIS

Deploying a WCF REST service on IIS Local machine (optional)

Adding Basic authentication

Securing basic authentication credentials using SSL over Http i.e. (Https)

The Basic authentication information is usually sent in plain text (encrypted by Base64 string which can be easily decrypted) over the network. Most of the browsers issue a warning about this secure information being sent as plain text. If you try to launch the API call in IE browser you'll see a warning something like below:



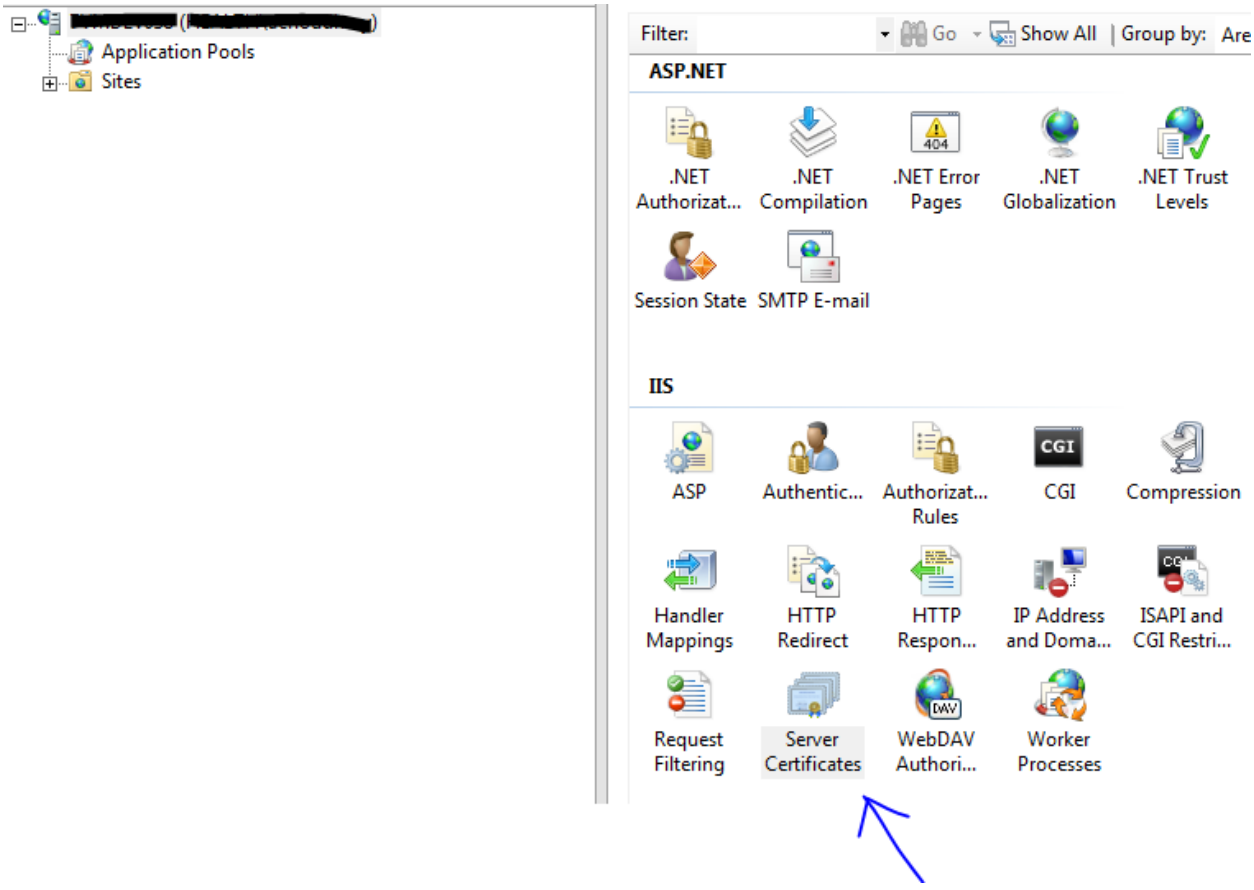
Here comes the Transport layer security into picture. All communication being done over the network can be secured using the Transport layer security by using SSL(Secure Sockets Layer) certificates.
To apply SSL security first you need a certificate. In order to get a real certificate one can go to certificates providers like Thawte, digicert, Godaddy etc. and purchase a certificate. These providers (not mentioning any specific provider but all in general) are trusted providers for issuing digital certificates to ensure that identity of certificate owner is verified.

But in development environment you don't need to purchase a SSL certificate in real. You can generate a local self-signed certificate from your machine. But the certificate would be only valid for your local use only. If you use the certificate over the network this would be invalidated. I'm not going into details so let's get started how to get a self-signed server certificate that you can use in development environment.

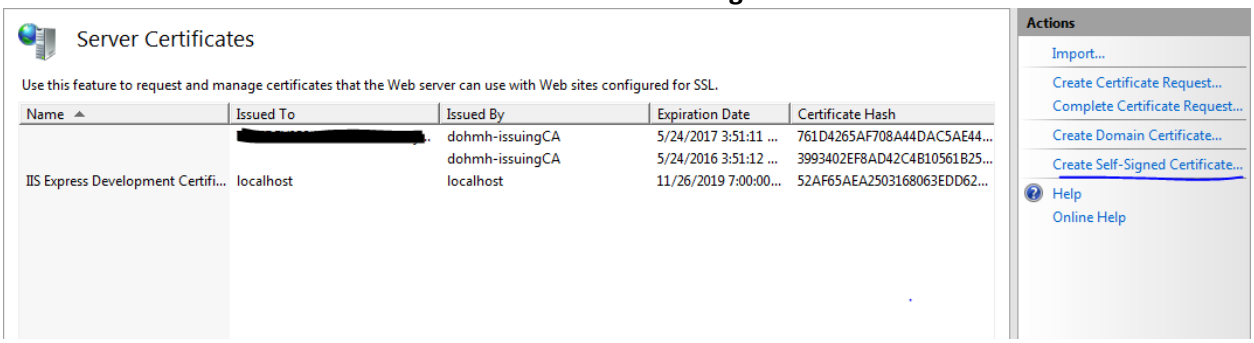
Creating a certificate and enabling the Https settings to use the certificate

You have two options either create a certificate using 'makecert' command. Here's a nice detailed article with quite good explanation about certificates. Or you can use IIS 7.0 or greater to create your certificate.

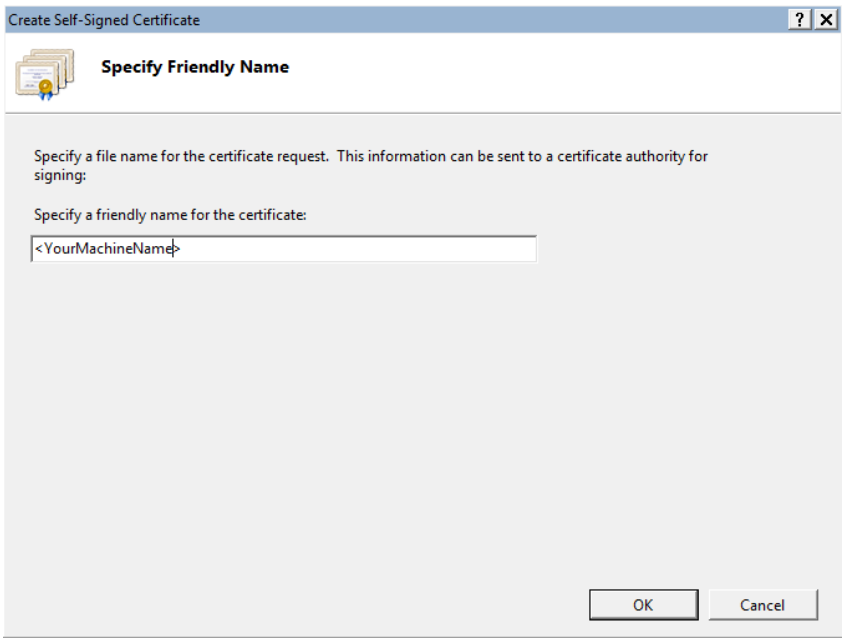
1. Launch Internet Manager from control panel or goto Run -> type '**Inetmgr'** -> press enter.
2. Select the ROOT node on left side panel. Look for `**Server Certificates**` under `**IIS**` on right panel.



3. Double click on '**Server Certificates**' and select **'Create self-signed certificate'**.

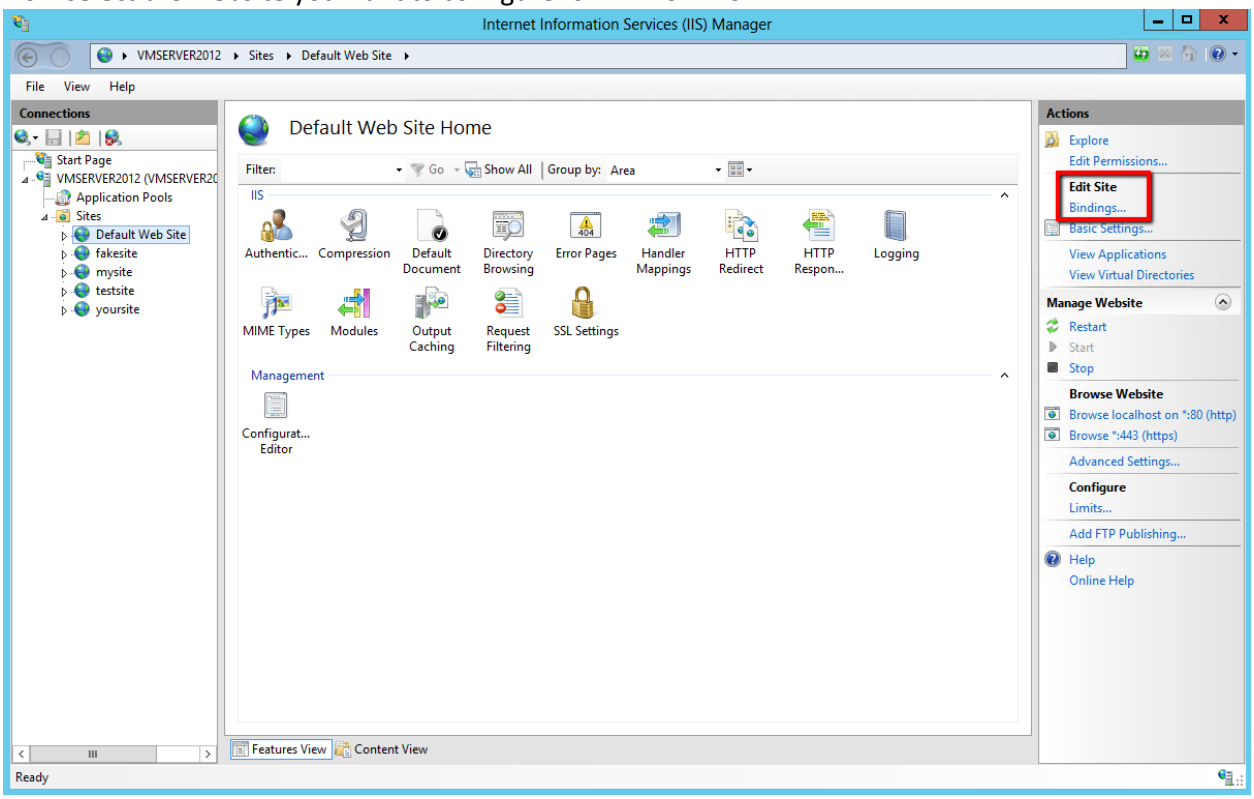


4. Now enter the name of certificate. This will be the friendly name of your certificate.

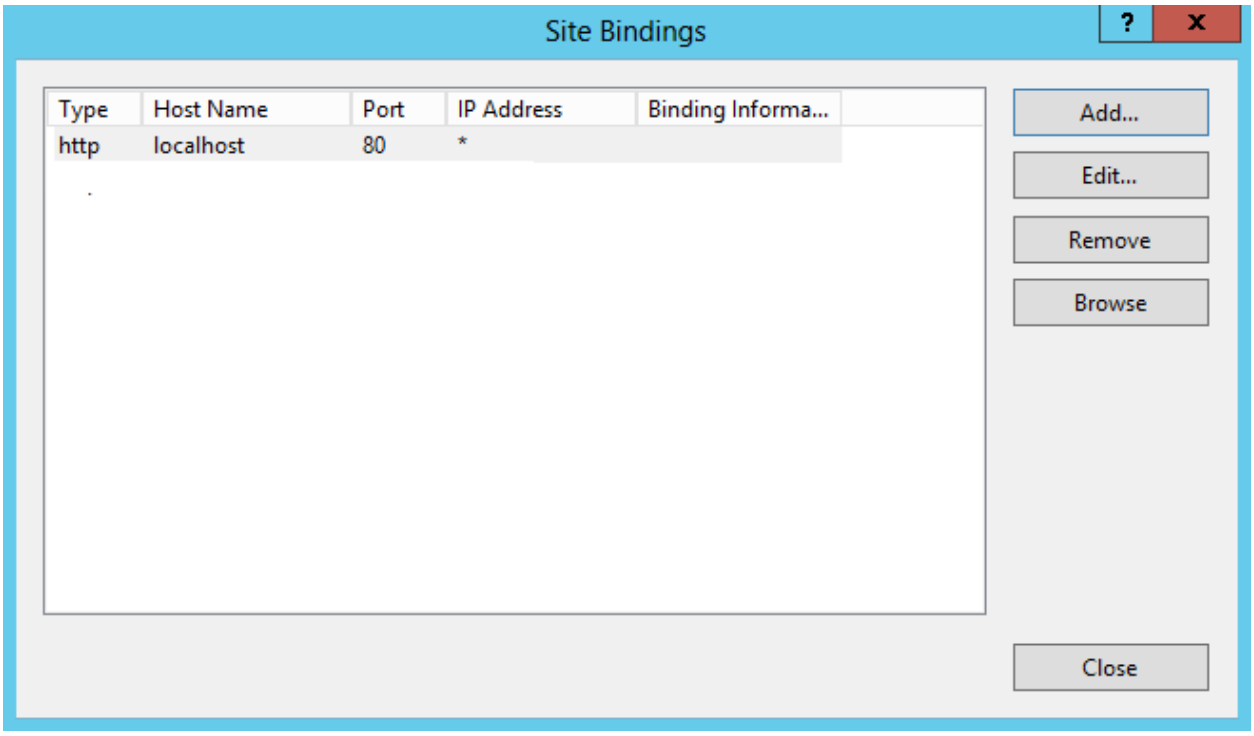


5. Click Ok and certificate that you have created would appear in parent window.

6. Now **select** the **website** you want to configure for HTTPS in IIS.

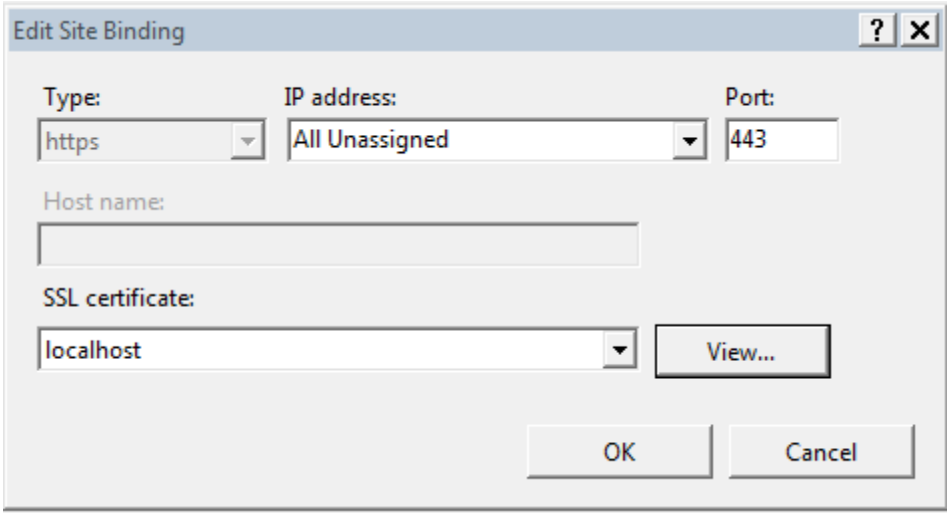


7. In the **Site Bindings** window, click **Add**.



8. In the **Add Site Binding** window, enter the following information:

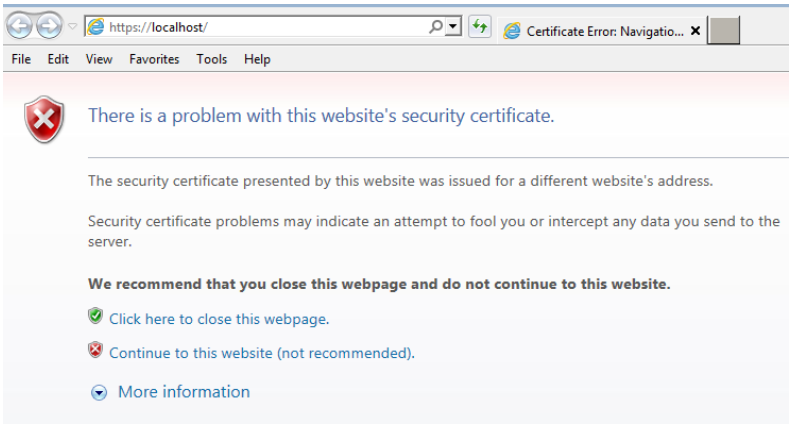
Type:	In the drop-down list, select **https**.
IP address:	In the drop-down list, select All Unassigned.
Port:	Enter 443. The port for SSL traffic is usually port 443.
SSL certificate:	In the drop-down list, select your recently imported SSL Certificate by its friendly name.



9. Click OK. Your SSL Certificate is now installed and the website is configured to accept secure connections.

Note: You may have to restart IIS or the server for it to recognize the new certificate (in some cases).

10. To test the setting open the site binding again and select Https binding and click on **'Browser'**. You might see below error.



11. To get rid of this error use the machine name exactly same as your certificate section "Issued to" says. E.g. If you open your certificate then you'll see **issued to** property and which should be your Machine name. If your machine is part of a domain then machine name would be like <machinename>.<xyz>.<domain> etc. so if you open it in your browser will fully qualified name of your machine then you won't be getting that error.

Setting up WCF REST service to use SSL (Https)

To enable the SSL over Http protocol follow below steps:

1. Add Protocol mapping for the webhttpbinding to use https.

```
<protocolMapping>
       <add binding="webHttpBinding" scheme="https"/>
</protocolMapping>
```

2. Under **binding** -> **webhttpbindings** enable security mode **Transport**.

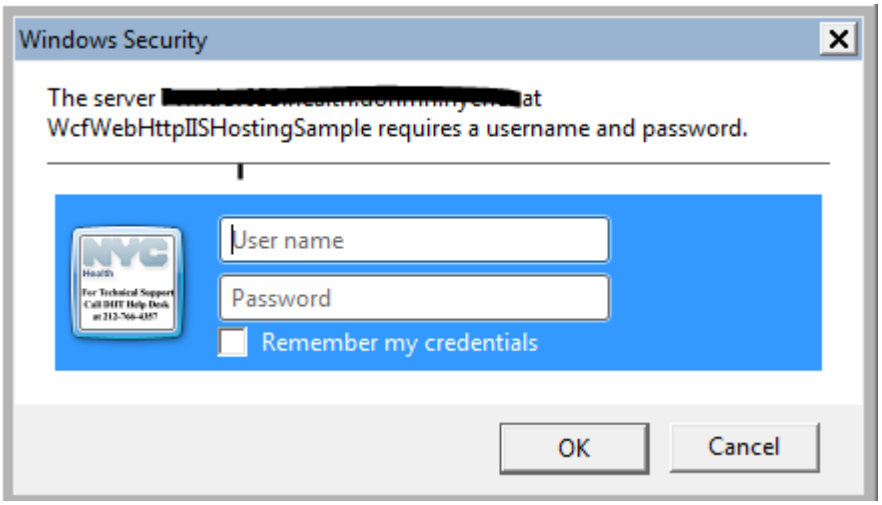
```
<bindings>
      <webHttpBinding>
         <security mode="Transport" />
        </binding>
      </webHttpBinding>
 </bindings>
```

3. Add a host base address under services -> service.

```
<service name="WcfWebHttpIISHostingSample.TestService"
behaviorConfiguration="ServiceBehavior">
     <host>
      <baseAddresses>
        <add baseAddress="https://desktop-pc.mydomain/WcfWebHttpIISHostingSample/api/"/>
      </baseAddresses>
     </host>
       …
       …

</services>
```

4. Now rebuild the service and test it in the browser. You will not see the warning of in-secure server communication any more.



I hope this article will help you. Feel free to like and share. The complete sample used in this Guide can be found here on Github.