

Re: Why is .NET CF 2.0 (HttpRequest Class) using 40-bit Encryption over HTTPS?

Re: Why is .NET CF 2.0 (HttpRequest Class) using 40-bit Encryption over HTTPS?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.compactframework/2008-06/msg000>

- *From:* "Greg Hellem" <ghellem@xxxxxxxxxxxx>
 - *Date:* Thu, 5 Jun 2008 17:14:49 -0500
-

I'm still debugging this issue and am now encountering a new/different symptom:

In an attempt to minimize the amount of code it takes to reproduce the issue that I'm encountering below (ArgumentOutOfRangeException) I am writing a small Smart Device Console application and applying the HttpRequest/Response classes as I do within the production mobile app that's giving me problems. However, while running the test app from within VS 2005, in debug mode, with the code executing on the device itself via ActiveSync I'm getting a WebException when trying to perform the EndGetResponse() method on the HttpRequest object. I'm not handling errors so I can easily see when/where problems occur.

The URL that I am using is formed as "<https://subdomain.domain.com>, allows only 128-bit encryption during communications and resides on our corporate intranet. Once again, I'm only getting this error when communicating to web sites that only allow 128-bit encryption. If the site allows 40 or 56-bit encryption, or requires no encryption at all, then my code works perfectly.

When executing the following statement in my subroutine GetResponse(ByVal ar As IAsyncResult):

```
Dim rs As HttpResponseMessage = CType(rq.EndGetResponse(ar), HttpResponseMessage)  
<== E R R O R ==>
```

WebException of: System.Net.WebException was unhandled
Message="Unable to read data from the transport connection."

StackTrace:

```
at System.Net.HttpWebRequest.finishGetResponse()  
at System.Net.HttpWebRequest.asyncGetResponse()  
at WorkItem.doWork()  
at System.Threading.Timer.ring()
```

Anyone have any insight as to the cause of this exception?

Thanks again for any help.

Greg Hellem

"Greg Hellem" <ghellem@xxxxxxxxxxxx> wrote in message
news:Pb2dnd83HaZklNXVnZ2dnUVZ v inZ2d@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Re: Why is .NET CF 2.0 (HttpWebRequest Class) using 40-bit Encryption over HTTPS?

This post is a continuation of these previous posts:

How do I install a SSL Certificate on a WinCE 5.0 Device? (Apr. 10, 2008)
Packaging setup.xml (for SSL Cert) into WinCE 5.0 CAB using CabWiz?
(Apr. 28, 2008)

Ladies and Gentlemen,

I am investigating how to properly implement SSL Certificates because our WinCE 5.0 mobile device are apparently using 40-bit encryption when communicating to our SSL Servers. I'm trying to determine why the encryption level is not 128-bit. Our IT Department increased the security levels on our servers to 128-bit and all of a sudden my .NET 2.0 CF apps were unable to communicate with the web apps until they lowered it back down to 40-bit!!!?

I had the IT guys setup a server on our intranet, with our SSL Cert installed and the security level raised to allow only 128-bit encryption, just so I can test my mobile application against it. I have converted and installed our SSL Cert on the mobile device (see previous posts listed above) and it still fails to communicate with the newly configured server. If they lower the security level down to 40-bit my application works perfectly.

According to the article listed here

(<http://msdn.microsoft.com/en-us/library/bb738067.aspx>):

"The .NET Compact Framework has built-in support for SSL. This support includes the use of sockets, Web (HTTP) requests, and XML Web service calls. You simply have to change the URL to begin with https:// instead of http:// to enable SSL."

It also goes on to state "Even if a certificate is not installed on the device, it is still possible to use SSL with any Web server that has a certificate installed."

Even though this article is referring to Windows Mobile 6 I have seen posts elsewhere indicating the same for WinCE 5.0 using .NET CF 2.0. I have followed all the "rules" for implementing HTTPS communications using the HttpWebRequest class by creating a CheckValidationResult method, in a LocalCertificatePolicy class, which always returns true and I specify "https://" in the URL when creating the HttpWebRequest object. Despite all of these efforts I cannot get my application to communicate at 128-bit encryption using these Symbol MC3090 WinCE 5.0 mobile devices!!!

What am I doing wrong??? Are there any properties of the created HttpWebRequest object that I have to set? I had run into a bug in the distant past whereby specifying the httpRequest.ContentLength when performing a "POST" caused problems when using HTTPS in .NET CF 1.0 but I removed that restriction when I moved the project to .NET CF 2.0. Could that still be an issue in 2.0? However, come to think of it, all I'm doing is a "GET" and I still have problems...

The following shows how I create the HttpWebRequest object and the properties I set:

Re: Why is .NET CF 2.0 (HttpWebRequest Class) using 40-bit Encryption over HTTPS?

Re: Why is .NET CF 2.0 (HttpRequest Class) using 40-bit Encryption over HTTPS?

```
hwrRequest = CType(WebRequest.Create("https://subdomain.domain.com"),  
HttpRequest)  
With hwrRequest  
    .AllowAutoRedirect = False 'Do not allow redirects  
    .KeepAlive = True 'Keep the connection alive  
    .Timeout = 120000 'Default milliseconds to wait  
    .AllowWriteStreamBuffering = True 'Allow data to be buffered  
End With  
ServicePointManager.CertificatePolicy = New LocalCertificatePolicy
```

For the "GET" operation I also set the "Content-Type" equal to
"text/html".

I then start a separate thread which performs:

```
Dim asncResult As IAsyncResult = hwrRequest.BeginGetResponse(AddressOf  
RespCallback, hwrRequest)
```

After that I simply wait for a response to the request...

I receive "ArgumentOutOfRangeException" within the routine that gets the
response when the code starts to read from the response stream buffer.
After researching this error I found that it can be due to a zero-length
buffer in the response stream, which apparently in the underlying
structure of the HttpRequest/HttpResponse classes contains a
negative value for the buffer length and this generates the
"ArgumentOutOfRangeException". However!! If I have the IT guys lower the
security level to 40-bit everything works again!???

In the initial (first) web request, all I'm doing is asking for the
contents of the main webpage on the specified URL. I'm not trying to do
anything fancy; I'm not downloading any resources or streaming anything
like sound or video. I just request the main webpage.

Single question: How do I properly implement 128-bit encryption using .NET
CF 2.0 and the HttpRequest/Response classes?

(And yes, I can successfully browse to the URL that I'm specifying in my
application using Internet Explorer on the mobile device, so I know it
must be supporting 128-bit encryption.)

I'm at the end of my rope...
PLEASE HELP!!!

Thank you,
Greg Hellem