ColdFusion Protocol Tags CFHTTP, CFINVOKE, CFLDAP support SSLv2

Posted At : November 14, 2006 3:12 PM | Posted By : Steven Erat
Related Categories: Java, Adobe, Events, Learning, ColdFusion, Macromedia

My reply to a comment to an earlier blog entry about **importing SSL certificates into ColdFusion cacerts file for CFHTTP** purposes warrants its own blog entry here as a separate topic.

The comment:

> [cfhttp url="https://httpmailbox1.beta.etrac.net/submit-to-etra... method="post" port="xxx" proxyserver="xxxx" proxyport="xxx" >
>
> I am having the same problem but I am using BlueDragon and CFMX. I am trying to connect to vendor using CFHTTP to send a XML file. The vendor keeps telling me that I am faliing the SSL handshake on his side. I was told by him that I needed to purchase a certificate form a trusted third party which we did (Verisign). This certificate was installed by my server team but I am refused connection at the vendor.
>
> Do I need to export the vendors certificate and install it on my WebLogic server using the Keytool.

It sounds as if the vendor is requiring SSLv3 with client authentication, rather than SSLv2 with only server authentication. The **documentation here** describes the conditions where you may have to import a certificate into ColdFusion for SSLv2 for server authentication, but this is often confused with the requirement for client auth:

> To use HTTPS with the cfhttp tag, you might need to manually import the certificate for each web server into the keystore for the JRE that ColdFusion uses. This procedure should not be necessary if the certificate is signed (issued) by an authority that the JSSE (Java Secure Sockets Extension) recognizes (for example, Verisign); that is, if the signing authority is in the cacerts already. However, you might need to use the procedure if you are issuing SSL (secure sockets layer) certificates yourself.

Lets back up a moment to consider the practical difference between SSLv2 and v3. First, imagine a simple HTTPS connection between a browser and server. A user at a browser types in the URL of a website beginning with https:// and the brower makes the request. Lets assume that the server is using SSLv2. The request gets to the server and the server replies with a message header stating it supports SSLv2 and sends its certificate. The browser receives the SSL certificate, inspects it, and negotiates a session key to be used for the remainder of the request/repsonse communication. This negotiation period is known as the SSL handshake.

This handshake or session key negotiation is done with asymmetric key cryptography,

and when a session key is agreed on by the browser and the client they switch to symmetric key cryptography. All this fuss to negotiate a session key is because symmetric key cryptography is much faster than asymmetric.

Anyway, the SSLv2 handshake does not require that the browser send its own certificate back to the server. In SSLv2 only the server has to prove its identity to the client.

However, in SSLv3 not only does the server have to prove its identity to the client, but the client (the browser in this case) also has to identify itself to to the server. The browser does that by sending its own certificate in return. So there are two certifcates involved in SSLv3, one on the server and one on the client.

In the case of a CFHTTP connection, the client is ColdFusion and the server is the target url such as https://httpmailbox1.... So if the vendor says that you need to send a certificate then the vendor is indirectly telling you that they support SSLv3 rather than SSLv2.

This conclusion is bad news unfortunately. ColdFusion MX 7 does not yet support SSLv3 with client authentication for its protocol tags such as CFHTTP, CFINVOKE, CFLDAP, etc.

Importing your new Verisign certificate into ColdFusion's cacerts file will do nothing for your problem. The cacerts file under ColdFusion is which Certificate Authorities ColdFusion will trust for the other end of the CFHTTP connection such as the scenario above for SSLv2. In order to support SSLv3 client authentication ColdFusion would have to perform an HTTP Request, get the HTTP Reply from the SSL website including the other end's SSL certificate, and then ColdFusion would have to send its own certificate back to the other server to say "Hey, I'm such and such ColdFusion server here, really I am". But ColdFusion does not yet do that.

You may need to seek a third party solution, perhaps a Java custom extension that did SSLv3 and HTTP for you, and you could integrate that into your ColdFusion application.

This problem is hinted at in a ColdFusion Technote on **Configuring Secure SSL Connection with LDAP Directory Server**:

> Also, cfldap does not support SSL V3 client authentication (user certificate authentication) in ColdFusion MX, as it did in ColdFusion 5. It only supports SSL V2 (basic username/password authentication over SSL).