Web Service SSL Issue: Trouble with Client Based Certificates

Posted At : August 27, 2010 2:28 PM | Posted By : Mark Kruger
Related Categories: Coldfusion Troubleshooting

I was recently brought in at the 11th hour to an issue where a web service worked fine on an older 32 bit install of ColdFusion, but was *not* working on the new 64bit install. All the keystore certs were up to date and the client was scratching their collective heads. The web service in question used SSL and presented a client certificate. Now in order to do this in CF you have to manually construct your SOAP request and pass it as XML using CFHTTP. This is because only CFHTTP has an attribute for clientcert and clientcertpassword (as of CF 8 I believe).

In any case, the code our customer was using was straightforward. It assembled a soap request and then issued a call to CFHTTP like so:

```
<cfhttp url="https://blahblah.com/?wsdl"
            charset="utf-8"
            port="443"
            method="post"
            result="response"
            clientcert="c:\somepath\somecert.p12"
            clientcertpassword="*some password*">

    <cfhttpparam type="xml" value="#soapPacketStuff#" />

</cfhttp>
```

On the "old" Coldfusion 8 "standard" server this worked fine. But on the brand spanking new, freshly patched and upgraded ColdFusion 8 enterprise server it was failing.

The Issue

The problem here is that a vulnerability in TSL (that's "SSL version 3.1" - see my previous post to learn more about the various SSL versions) was discovered in fall of 2009. This vulnerability allowed a "man in the middle" attacker to force a "renegotiation" of the handshake - thus allowing the insertion of arbitrary code into the request. Obviously this is a serious flaw. The *fix* was for vendors to simply disable the renegotiation feature in their products. So, for example, IIS 7 does not allow renegotiation by default.

So why is this issue not out there in the CF blogging world yet? First, I think that the use of client certs is a fairly small universe of ColdFusion applications. Second, Sun followed suit and *fixed* this issue with version 1.6.0_19 by disabling renegotiation. I think that many CF Servers are still using 1.6.0_18b or below - so this issue has yet to really rear its ugly head.

In any case, regular SSL requests continue to work as always with renegotiations disabled. A client certificate driven request is different however. It often *requires* renegotiation because of the complexity of the handshake (with 2 certificate pairs involved. That means *client certificate driven CFHTTP SSL requests using the 1.6.0_19+ JVM will often fail to successfully negotiate a secure session.* I want to say they will *probably* or *certainly* fail, but I'm not positive on that score.

The Fix - Such as it is

You have two ways of fixing this. You can roll back to 1.6.0_18. Seeing as how the current build is _20 you may not want to do this. Or, if you want to stay on 1.6.0_19+ you can add the following argument to your JVM.config file or files.

```
-Dsun.security.ssl.allowUnsafeRenegotiation=true
```

Obviously this allows for the server to renegotiate successfully. Your JVM will be vulnerable to the TLS exploit, but depending on your situation it may be a fairly low risk proposition. Still, you should take it into account. Consider running a separate JVM instance just for your web service. Some edge security devices can sniff out this issue as well.

Many thanks to my good friend and troubleshooting colleague Vlad Friedman from the **Edge Web** for figuring this out. I would also recommend that your read Chris Mahns blog entry on **TLS Remediation**.