SSL and the trusted keystore in Java

Posted At : January 29, 2005 10:58 AM | Posted By : Mark Kruger
Related Categories: Coldfusion Troubleshooting

This tip is from my old blog - but it is worth repeating. The Java runtime must trust a cert to get it to work with it properly.

These notes are the result of solving a particularly tricky problem with webservices on CF Talk. This helpful "keystore" procedure came from the diligent investigation of Mike Chambers and Trevor Baker.

There is a tricky nuance to using with SSL in CFMX. In order to make an outgoing SSL request, the requesting agent must first obtain the "public" key. This public key is available from a "trusted certificate authority". Verisign, Thawt, and equifax are 3 well-known "trusted authorities". In your browser, if the certificate is not a "trusted athority", a warning message informs you that, while the cert may be good in other ways (not expired etc.) it is not from a source you have listed as trusted. If you choose, you can simply accept the certificate anyway. Note, encryption is determined by the type and size of the key - not by whether the authority is trusted or not. All things being equal, a certificate from a non-trusted authority will result in the same level of protection as that from a trusted authority.

In CF 5 an outgoing SSL request using was successfully negotiated if the cert was found in the "root certificate store" of the server. In other words, if you had indicated the cert was trusted and allowed the key to be installed, CF 5 was able to use it. In CFMX however, the Java Run-time is unaware of the root certificate store. Instead, it has it's own cache of "trusted authorities" and it installs certs as needed based on this cache.

To discover the list of trusted authorities in your Java run time, try the following command line code:

```
C:\CFusionMX\runtime\jre\lib>keytool -list -storepass changeit -noprompt -keystore
C:\CFusionMX\runtime\jre\lib\security\cacerts
```

Of course you will want to change the path to your CFMX runtime directory. When I run this command on my CFMX dev box with the standard 1.3 JRE I get the following:

```
Keystore type: jks
Keystore provider: SUN

Your keystore contains 10 entries

thawtepersonalfreemailca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): 1E:74:C3:86:3C:0C:35:C5:3E:C2:7F:EF:3C:AA:3C:D9
thawtepersonalbasicca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): E6:0B:D2:C9:CA:2D:88:DB:1A:71:0E:4B:78:EB:02:41
verisignclass3ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 78:2A:02:DF:DB:2E:14:D5:A7:5F:0A:DF:B6:8E:9C:5D
thawtepersonalpremiumca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): 3A:B2:DE:22:9A:20:93:49:F9:ED:C8:D2:8A:E7:68:0D
thawteserverca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): C5:70:C4:A2:ED:53:78:0C:C8:10:53:81:64:CB:D0:1D
verisignclass4ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 1B:D1:AD:17:8B:7F:22:13:24:F5:26:E2:5D:4E:B9:10
verisignclass1ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 51:86:E8:1F:BC:B1:C3:71:B5:18:10:DB:5F:DC:F6:20
```

```
verisignserverca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 74:7B:82:03:43:F0:00:9E:6B:B3:EC:47:BF:85:A5:93
thawtepremiumserverca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): 06:9F:69:79:16:66:90:02:1B:8C:8C:A2:C3:07:6F:3A
verisignclass2ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): EC:40:7D:2B:76:52:67:05:2C:EA:F2:3A:4F:65:F0:D8
```

Obviously I trust Thawte and Versign certificates.

What happens when I need to make an SSL request to a site using a cert that is not from one of these authorities? The request will fail. For it to succeed, I must tell my JRE to "trust" the authority in question. To do that you will need to use the keytool to "import" the cert from the authority in question. For example, if you wanted to import the "instant SSL" certificate, you would need to import the 2 signing certificates they use to create their own certs. They use the following 2 certs:

GTE CyberTrust Root CA
Comodo Class 3 Security Services CA

Both of these certs are available through links at **instant SSL installation support**. Save each of them into a text file - cert1.crt and cert2.crt (or whatever), then use the key tool to import them into your store. Here is the command line syntax:

```
C:\CFusionMX\runtime\jre\lib>keytool -import -keystore c:\CFusionMx\runtime\jre\
lib\security\cacerts -alias instantssl -storepass changeit -noprompt -trustcacer
ts -file c:\temp\cert1.crt
```

More information regarding the "keytool" is available at **Sun Keytool docs**. Once this code is run, you can re-run the initial command line code and look at your entries again. You should see something like this:

```
instantssl, Dec 5, 2002, trustedCertEntry,
Certificate fingerprint (MD5): C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
```

Now, when your outgoing request negotiates the connection it will "trust" the public key provided to complete the SSL negotiation.