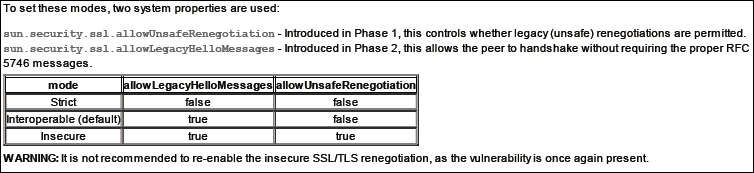
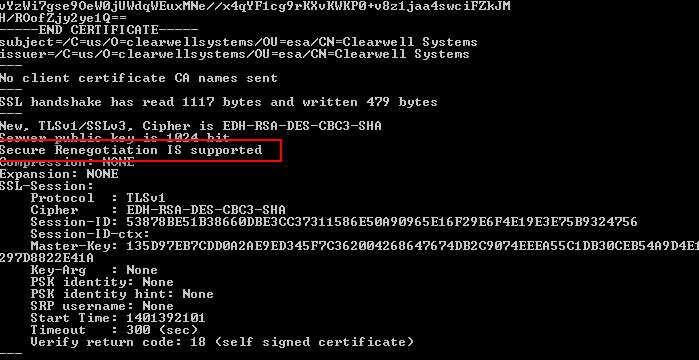
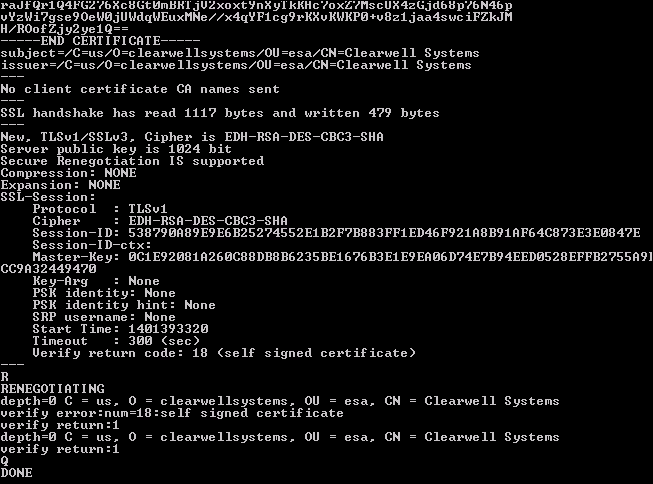
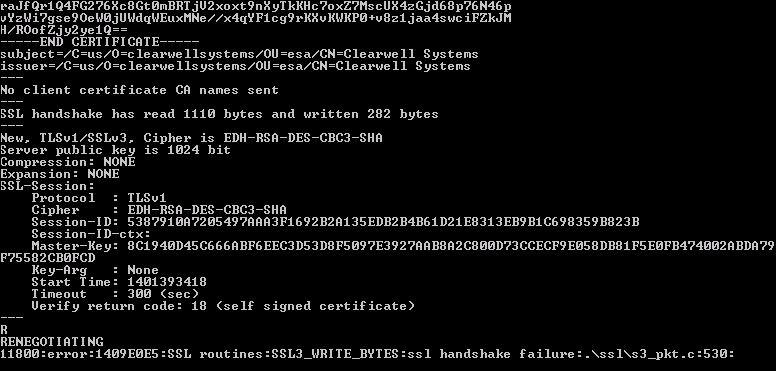
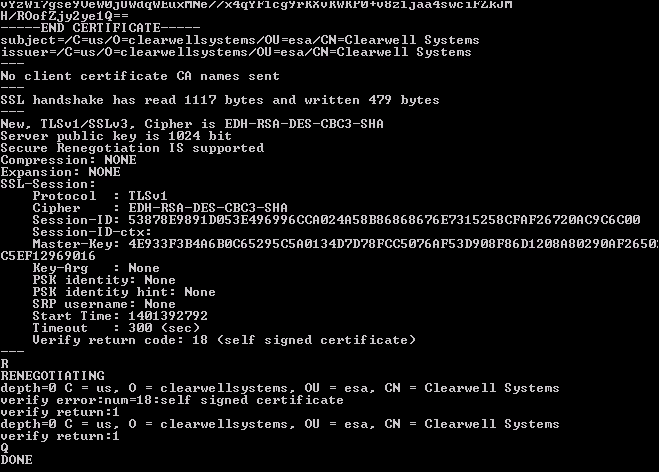
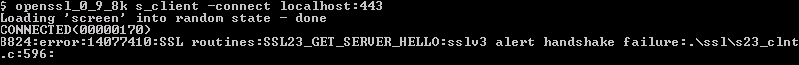
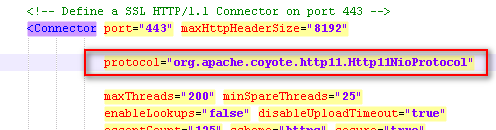
**Testing Oracle Jrockit ‘operation modes’ for handling the TLS MITM renegotiation issue (CVE-2009-3555)**JeremyC 29/5/2014

From <http://www.oracle.com/technetwork/java/javase/documentation/tlsreadme2-176330.html>

As per the above link, this was resolved in Java 1.6.0\_22, where RFC5746 was implemented.  
  
To handle clients that still don’t talk RFC5746, three operation modes can configured using two Java system properties…  
  


So the default is the following:  
-Dsun.security.ssl.allowUnsafeRenegotiation=false  
-Dsun.security.ssl.allowLegacyHelloMessage=true  
  
*The Phase 1 Fix was to disable renegotiations by default until a RFC 5746-compliant fix could be developed. Renegotiations could be reenabled by setting the sun.security.ssl.allowUnsafeRenegotiation system property.***Note**: I believe this setting of “false” is why the ssllabs.com/ssltest returns the following success:  
  
You can also see this by using openssl.exe:  
  
…  
 *The Phase 2 fix uses the same systemproperty, with the addition of the sun.security.ssl.allowLegacyHelloMessage system property to require the use of RFC 5746 messages.*

**Note**: I believe the setting of “true” is why the ssllabs.com/ssltest report returns the following failure:  
  
**Note**: Therefore, I strongly suspect that setting “-Dsun.security.ssl.allowLegacyHelloMessage=false” (i.e. configuring strict mode) will resolve this faliure.  
  
**Note**: Testing this property change is a little tricky however, if you don’t have an appliance open to ssllabs.com/ssltest. Testing with openssl.exe is also tricky, because you need to use a version of 0.9.8k (or earlier). Versions after this include the implementation of RFC5746, so you cannot use such as version to confirm that the conncection fails if the server JVM uses ““-Dsun.security.ssl.allowLegacyHelloMessage=false”. In the end, I had to rebuild my own 0.9.8k to test this (I could not find a download of the version on the Internet).  
  
**TEST#1:**  
Start the server in default mode (i.e. neither of the two properties explicitly set). Remember, the default values are:  
sun.security.ssl.allowUnsafeRenegotiation=false  
sun.security.ssl.allowLegacyHelloMessage=true  
*Testing with OpenSSL 1.0.1e:*  
  
…  
  
**Note**: This version of the OpenSSL client **does** talk RFC5746. The initial connection is successful, and client-initiated connection renegotiation (the “R” command shown above) is also successful. All is good – apart form the fact that, because renegotiation is possible, a DoS issue is possible – hence ( I think) the cause of the following ssllabs.com/ssltest failure:  
  
  
*Testing with OpenSSL 0.9.8k:*  
  
…  
  
**Note**: This version of the OpenSSL client does **not** talk RFC5746, but the server is being permissive (because “sun.security.ssl.allowLegacyHelloMessage=true “) by allowing the initial connection. However, the client-initiated connection renegotiation fails (because “sun.security.ssl.allowUnsafeRenegotiation=false”). I \*think\* this might explain the following ssllabs.com/ssltest failure:  
  
(I’m not 100% sure of this however, since we can clearly see the renegotiation failed. Perhaps the ssllabs.com test is just checking for an initial handshake faliure – see TEST#2 below).  
 **TEST#2:**  
Set ““sun.security.ssl.allowLegacyHelloMessage=false”, in order to start the server in strict mode. Windows Task Manager confirms the two property values I have explicitly set:  
  
*Testing with OpenSSL 1.0.1e:*  
  
…  
  
**Note**: This version of the OpenSSL client **does** talk RFC5746, so, again, the initial connection is successful and renegotiation is successful (the “R” command shown above).  
  
*Testing with OpenSSL 0.9.8k:*  
  
**Note**: This version of the OpenSSL client does **not** talk RFC5746. Since we now have the server in ‘strict’ mode (because “sun.security.ssl.allowLegacyHelloMessage=true)”, the initial connection is refused.   
I \*think\* this must be what ssllabs.com/ssltest is looking for to pass the following test:  
  
(But it’s hard to confirm this without an appliance open to the Internet to point ssllabs.com at).  
  
**TEST#3:**  
Using the default properties again, but this time with renegotiation disabled in Tomcat server.xml :  
  
*Testing with OpenSSL 1.0.1e:*Connection is successful. Renegotiation fails.  


*Testing with OpenSSL 0.9.8k:*Connection is successful. Renegotiation fails.  
  
**(Note: The possible implications of doing this needs to be confirmed as safe).**   
**Conclusions:**  
o I suspect that setting “sun.security.ssl.allowLegacyHelloMessage=false” will resolve the following report failure:  
  
(I think the test is only checking if the initial connection fails or not. If the connection succeeds then the test fails. Seems fair, since the test tool likely has no notion of the Oracle JVM’s special property “sun.security.ssl.allowLegacyHelloMessage”. And I bet the test tool doesn’t do what my test is doing – i.e. connect then try a renegotiation). **I guess I should be able to confirm this with a Wireshark capture.**  
  
o I also suspect that disabling server-side renegotiation completely is the only way to resolve the following DoS report failure:  
  
  
**TODO:**  
1. Get confirmation that setting “sun.security.ssl.allowLegacyHelloMessage=false” is safe.

2. Get confirmation of the steps to configure Tomcat server.xml to completely disable renegotiations and confirm that this is safe, including in a clustered environment.  
  
**END**