Common problems while configuring SSL in WebSphere Message Broker v8

Rao Nanduri (rnanduri@us.ibm.com),
Message Broker Support Team
Lisbeth Arriaza (larriaza@us.ibm.com),
Java Security Support Team
4 April 2013
Agenda

- SSL Security introduction
- SSL in JMS and TCPIP nodes
- SSL in WebServices
- Common SSL Problems (1-6)
- Product Defect APARS
- Summary
SSL Protocol

SSL allows two parties to communicate securely by:

- Providing authentication
- Encrypting the communication
- Providing data integrity

The SSL protocol functionality is provided by the IBM Java Secure Socket Extension (JSSE) component in the IBM JVM.

Message Broker uses the JSSE to provide its SSL support.
SSL protocol

- **Keystore.** The Keystore database is a repository of certificates. It contains the private key of the server.

- **Truststore.** The trust store database is also a repository of certificates but it does not contain private keys. It only contains public keys of the trusted entities.

- The only supported format of the store files in Message Broker is the Java keystore (JKS) format.
SSL cipher suites

- Along with truststores and keystores, the transport level SSL requires the configuration of protocols, algorithms and ciphers.

- CipherSuites – Set of algorithms providing means of Encryption.
SSL handshake

SSL Messages

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Client hello</td>
<td>2. Server hello</td>
</tr>
<tr>
<td>5. Server key exchange  optional</td>
<td>6. Server hello done</td>
</tr>
<tr>
<td>7. Certificate  optional</td>
<td></td>
</tr>
<tr>
<td>8. Client key exchange</td>
<td></td>
</tr>
<tr>
<td>9. Certificate verify  optional</td>
<td></td>
</tr>
<tr>
<td>10. Change cipher spec</td>
<td>12. Change cipher spec</td>
</tr>
<tr>
<td>11. Finished</td>
<td>13. Finished</td>
</tr>
<tr>
<td>15. Close messages</td>
<td>15. Close messages</td>
</tr>
</tbody>
</table>

SSL in Java Message Service (JMS)

- WMB v8 Supports JMS 1.1 Specification
  - that does not support controlling or configuring message integrity and message privacy.

- Broker environment can be configured to work with those JMS providers that support JMS Clients using SSL protocol.

- SSL configuration is JMS Provider dependent.
SSL in Java Message Service (JMS)

1. No SSL related properties defined in JMS nodes

2. The transparent connection possibilities are
   - Defining them via Environment variable
     set SSL_CONFIG=
     -Djavax.net.ssl.trustStore=%DummyClientTrustFile.jks
     set SSL_CONFIG=
     -Djavax.net.ssl.trustStorePassword=<passwd>

3. Define in configurable service properties
SSL in TCPIP Nodes

mqsicreateconfigurableservice MB8BROKER
  -c TCPIPServer
  -o TCPIPServerService
  -n Port,SSLProtocol,SSLCiphers,SSLCipientAuth
  -v 1455,SSLv3,SSL_RSA_WITH_RC4_128_MD5;
    SSL_RSA_WITH_3DES_EDE_CBC_SHA

mqsicreateconfigurableservice MB8BROKER
  -c TCPIPClient
  -o TCPIPClientService
  -n Port,Hostname,SSLProtocol,SSLCiphers
  -v 1455=localhost,SSLv3,SSL_RSA_WITH_RC4_128_MD5;
    SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL in WebServices

- Key is obtained from a keystore and verifies it against a list of keys in a truststore.

- Properties are set using the mqsichangeproperties and mqsisetdbparms commands and verify them using mqsiexportproperties.

- Key/TrustStore files can be defined at broker or EG level for SOAP nodes.

- Execution Group Definitions take precedence over the broker level.
Debugging SSL connections

- IBM JSSE2 provides a flag to debug SSL connections: `-Djavax.net.debug=true`
  - In Broker environment, one can set by adding the following env variable to `<brokerInstall>/bin/mqsiprofile`:
    ```bash
    Export IBM_JAVA_OPTIONS= Djavax.net.debug=true
    ```
- The JSSE debug messages are printed to `<Broker_WorkPath>/components/<Broker>/<EGUUUID>/stdout log file.`
Common SSL problems – Problem #1.

SSL client does not trust the SSL server.

- The following exception is thrown on the SSL client side logs:

```
javax.net.ssl.SSLHandshakeException: com.ibm.jsse2.util.j: PKIX path building failed:
java.security.cert.CertPathBuilderInterfaceException: unable to find valid certification path to requested target
```
...Problem #1

- Review the SystemOut log after enabling the JSSE debug flag.
  - Look for the certificate chain sent from the Server. For example:

```plaintext
*** Certificate chain
cert[0] = [
    [Version: V3
     Subject: CN=server1.outsourcing.local
     Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5
     Validity: [From: Mon Mar 04 07:59:48 CST 2013,
                  To: Tue Mar 04 07:59:48 CST 2014]
     Issuer: CN=server1.outsourcing.local
     SerialNumber: [1362405588]
    ]
]```
Problem #1

- Look for the truststore initialization.

Adding as trusted cert:
Subject: OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US
Issuer: OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US
Algorithm: RSA; Serial number: 0x3c9131cb1ff6d01b0e9ab8d044bf12be
Valid from Sun Jan 28 18:00:00 CST 1996 until Wed Aug 02 17:59:59 CST 2028

- In this case, the exception occurs, because the truststore does not contain the Server side certificate.
To resolve this problem, import the server certificate into the SSL client’s truststore.

For example:

```
keytool -import -trustcacerts -file servercert.arm -keystore clienttrust.jks
```
Common SSL problems – Problem #2

Mutual SSL authentication fails.
- The following exception is displayed on the SSL server logs:

```
javax.net.ssl.SSLHandshakeException: null cert chain
```
...Problem #2

- Review the SSL client SystemOut log after enabling the JSSE debug flag.

- Look for the CertificateRequest message sent by the server

  *** CertificateRequest
  Cert Types: RSA, DSS, ECDSA
  Cert Authorities:
  <CN=server1.outsourcing.local>
…Problem #2

- The CertificateRequest message contains a list of Certificate Authorities trusted by the SSL server.

- The SSL client needs to have a personal certificate issued by any of these certificate authorities.
...Problem #2

- To resolve the problem, create a personal certificate on the SSL client side keystore that is issued by any of the Certificate Authorities trusted by the server.
Common SSL problems – Problem #3

- Mutual SSL authentication fails. The certificate chain on the SSL client side is not complete. The following exception is displayed on the SSL server logs:

```java
javax.net.ssl.SSLHandshakeException: null cert chain
```
Problem #3

- Review the SSL client SystemOut log after enabling the JSSE debug flag
- Verify that you have a personal certificate on the keystore:

```
found key for : server1
chain [0] = [
  [  
    Version: V3
    Subject: CN=Server1.outsourcing.local, OU=Test, O=OrganizationTest, L=Guatemala, ST=Guatemala, C=GT
    Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

    Validity: [From: Mon Mar 04 11:00:19 CST 2013,
                To: Tue Mar 04 11:00:19 CST 2014]
    Issuer: CN=IntermediateCA, OU=Intermediate, O=Intermediate, ST=Guatemala, C=GT
    SerialNumber: [1]
  ]
```

***
Problem #3

- Validating the chain can also be performed by using the keytool utility:
  ```
  keytool -list -v -keystore clientkey.jks
  ```

  Alias name: server1
  Creation date: Mar 4, 2013
  Entry type: keyEntry
  **Certificate chain length: 1**
  Certificate[1]:
  Owner: CN=Server1.outsourcing.local, OU=Test, O=OrganizationTest, L=Guatemala, ST=Guatemala, C=GT
  Issuer: CN=IntermediateCA, OU=Intermediate, O=Intermediate, ST=Guatemala, C=GT
  Serial number: 1
  Valid from: 3/4/13 11:00 AM until: 3/4/14 11:00 AM
  Certificate fingerprints:
Problem #3

- Review the list of Certificate Authorities sent during the CertificateRequest message:

*** CertificateRequest
Cert Types: RSA, DSS, ECDSA
Cert Authorities:
<CN=TCIMOSGT27.outsourcing.local>
<CN=CATest, OU=CA, O=CATest, L=Guatemala, ST=Guatemala, C=GT>

As seen above, we don’t have a personal certificate in the client’s keystore that is issued by any of the above Certificate authorities.
...Problem #3

- In this example, we want to have the following chain:

  Chain[0]
  Owner: CN=Server1.outsourcing.local, OU=Test, O=OrganizationTest, L=Guatemala, ST=Guatemala, C=GT
  Issuer: CN=IntermediateCA, OU=Intermediate, O=Intermediate, ST=Guatemala, C=GT

  Chain[1]
  Owner: CN=IntermediateCA, OU=Intermediate, O=Intermediate, ST=Guatemala, C=GT
  Issuer: CN=CATest, OU=CA, O=CATest, L=Guatemala, ST=Guatemala, C=GT

  Chain[2]
  Owner: CN=CATest, OU=CA, O=CATest, L=Guatemala, ST=Guatemala, C=GT
  Issuer: CN=CATest, OU=CA, O=CATest, L=Guatemala, ST=Guatemala, C=GT

- As noted in the previous slide, we only have the first certificate in the chain.
- The SSL server trusts the certificate highlighted above.
Problem #3

To resolve the problem we have 2 options:

- Ask the Certificate Authority to fix the chain and to send us the correct certificate reply
- If we have all the certificates of the chain, we can complete the chain using the steps in the next slide.
…Problem #3

Fixing the chain

- Install all your certificates in your Internet Explorer browser.
  - Double click each .crt file
  - Click Install Certificate
  - Follow the “Certificate Import Wizard” (Leave all defaults)

- Follow steps 5-10 in the URL below:

  http://www-01.ibm.com/support/docview.wss?uid=swg21231482
Common SSL problems – Problem #4

- The JSSE trace shows the certificate_unknown SSL error with the following exception:
  javax.net.ssl.SSLHandshakeException: com.ibm.jsse2.util.g: Extended key usage does not permit use for TLS client authentication.

- error message indicates that the certificate is being for client authentication but the Extended Key Value indicates it can only be used for server authentication.
Problem #4

- In a Digital Certificate the "Extended key usage" further refines key usage extensions. An extended key is either critical or non-critical. If the extension is critical, the certificate must be used only for the indicated purpose or purposes. If the certificate is used for another purpose, it is in violation of the CA's policy.

- For a certificate to be marked for use for Server Authentication only, the Extended Key Usage Field in the certificate must be configured with the Critical flag set to True and the Value set to 1.3.6.1.5.5.7.3.1. For Client Auth, it is set to 1.3.6.1.5.5.7.3.2.
Problem #4

- In current scenario, the certificate used for mutual authentication from client.

```
[  
   Version: V3
   Subject: CN=was.ibm.com, OU=US O=NC,..
   Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5
   ...
   [9]: ObjectId: 2.5.29.37 Criticality=false
   ExtKeyUsage [ 1.3.6.1.5.5.7.3.1]
```

The flag is **not** set to **true** and the value indicates it is for **server authentication**.

**Solution:** Certificates should be re-generated setting it to true and use the client authentication value instead for using it from client side.
Common SSL problems – Problem #5

- The JSSE trace shows
  - java.io.IOException: SSL handshake failed. Cipher suite in SSL Session is SSL_NULL_WITH_NULL_NULL:java.io.SSL_NULL_WITH_NULL_NULL

- If no suitable cipher suites exists, or in some cases when broker doesn't support the specific CipherSuite, the SSL handshake could fail.

- If SSL V3 or TLSv1.0 ciphers are used, then SSL V2 support is disabled. And when SSL V2 ciphers are specified, then SSL V3 and TLSv1.0 support is disabled.
Common SSL problems – Problem #6

- Broker Logs reports the following error for a SOAPRequest node request:
  - BIP3165S: An error occurred whilst performing an SSL socket operation. Operation: 'connect'. Error Text: 'javax.net.ssl.SSLProtocolException: Server resumed session with wrong protocol version'.

- If the remote side socket attempts to resume an SSLSession on a different protocol version, then Message Broker rejects the handshake attempt.
- This could be due to privilege escalation attack at the network firewalls due to insufficient access.
Known WMB Product Defects in v8

- The following exception reported in some scenarios at fixpack level 8001 when client authentication is enabled, when Broker is used as client -
  - javax.net.ssl.SSLException:
    - java.lang.ArrayIndexOutOfBoundsException: Array index out of range: 0
- APAR IC88513 is raised to address this problem.
Summary

- Message Broker provides SSL functionality through JSSE2 component of the IBM JVM, for authentication, data integrity and encryption in its communication with a number its interfaces.

- JSSE2 debug facility would help troubleshooting the SSL connection problems.
Reference Material

- Message Broker Infocenter
  - http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/index.jsp
- Developer works
- Message Broker Support Page for updates
  - http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Message_Broker
- Toolkit for sample applications
- IBM® Message Broker Blog
Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/

- Join the Global WebSphere Community: http://www.webspherelusergroup.org

- Access key product show-me demos and tutorials by visiting IBM® Education Assistant: http://www.ibm.com/software/info/education/assistant


- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html
1. **Get notified on upcoming webcasts**  
   Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**  
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

3. **Be connected!**  
   Connect with us on Facebook  
   Connect with us on Twitter
Questions and Answers