Configure JALoP V2 TLS

1. On the machine running jald, create private/public key pair and create a public certificate to import on the subscriber

   ```
   cd ~/JALoP/test-input/certs
   openssl genrsa -out publisher.key 2048
   openssl req -new -key publisher.key -out publisher.csr
   openssl x509 -req -days 3650 -in publisher.csr -signkey publisher.key -out
   publisher.crt
   ```

2. On the JJNL subscriber server, create the JALoP keystore

   ```
   cd ~/jjnl/jnl_test/certs
   keytool -genkeypair -keyalg rsa -keystore server.jks -storepass changeit -
   alias <alias> -ext san=<domain-name>,ip:<ip-address>
   ```

   NOTE: If you are generating this keypair on a system running a Java 11 version later than 11.0.12, and you expect to connect to a peer running a Java 8 version older than 8.0.301, you must specify the additional parameter `–J-Dkeystore.pkcs12.legacy` as a part of the keytool parameter.

   The keystore password ‘changeit’ can be changed to the desired password.

   The alias can be any string that identifies this keypair.

   Either the domain name or the IP address needs to be filled in for the SAN.

   <domain-name> is the domain name of the server. For example, test-server.com. A domain-name only SAN would look like: san=dns:<domain-name>

   <ip-address> is the IP address of the server. For example, 127.0.0.1. An IP address only SAN would look like: san=ip:<ip-address>

3. Export the server’s public server certificate. Then, copy the certificate to the JALoP peer for mutual authentication.

   ```
   cd ~/jjnl/jnl_test/certs
   keytool -exportcert -rfc -keystore server.jks –alias <alias> > server.pem
   cp server.pem ~/JALoP/test-input/certs
   ```

4. Create a link to the copied server.pem certificate. If this is not done, the JALoP peer will give an “unknown_ca” error when trying to connect.

   ```
   cd ~/JALoP/test-input/certs
   openssl x509 -noout -hash -in server.pem
   ln server.pem <hash output from above command>.0
   ```

5. Update the following paths in the jald.cfg file, if necessary

   ```
   # the path to the private key, used for TLS negotiation
   private_key = ”/path/to/JALoP/test-input/certs/publisher.key”;

   # the path to the public cert, used for TLS negotiation
   public_cert = “/path/to/JALoP/test-input/certs/publisher.crt”;

   # directory containing the CA certificate(s) to use for TLS negotiation
   cert_dir = “/path/to/JALoP/test-input/certs”;
   ```

6. Add the peer’s certificate into the subscriber’s keystore.

   ```
   cd ~/jjnl/jnl_test/certs
   keytool -importcert -keystore server.jks -file /path/to/publisher.crt -alias <peer>
   ```

   The alias can be any string that identifies this certificate.

   Note: If you are intending to use multiple keystores (one for server certificates and one for remote certificates), place this file into the remote certificate keystore. If you are generating this keypair on a system running a Java 11 version later than 11.0.12, and you expect to connect to a peer running a Java 8 version older than 8.0.301, you must specify the additional parameter –J-Dkeystore.pkcs12.legacy as a part of the keytool parameter.

7. Update the following items in the sampleHttpSubscriber.json file, if necessary

   ```
   configureTLS: “on”,
   ```

   Keystore only:

   ```
   “Key Store Passphrase”: “changeit”,

   “Key Store”: “/path/to/jjnl/jnl_test/certs/server.jks”,
   ```

   Keystore and Trust Store:

   ```
   “Key Store Passphrase”: “changeit”,

   “Key Store”: “/path/to/jjnl/jnl_test/certs/trust_store/server.jks”,

   “Trust Store Passphrase”: “changeit”,
   ```

“Trust Store”: “/path/to/jnl/jnl_test/certs/trust_store/remotes.jks”

8. Start jald using the modified cfg file

```
cd ~/JALoP
jald –no-daemon –c </path/to/JALoP/jald-modified.cfg>
```

9. Start the JJNL subscriber to connect to jald using the modified JSON file

```
cd ~/jjnl/jnl_test
java –jar target/jnl_test-2.x.x.x.jar /path/to/sampleHttpSubscriber-
modified.json
```