Configure JALoP V1 TLS

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

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

2. On jjnl subscriber server, create JALoP keystore

   ```
   cd ~/jjnl/jnl_test/certs
   sudo keytool -genkeypair -keyalg rsa -keystore server.jks -noprompt -storepass changeit -keypass changeit -dname "CN=test, OU=ID,O=test, L=test, S=MD, C=US"
   ```

3. Export the public server certificate (this will need to be imported on the remote JALoP peer for mutual authentication) Copy to the jald certs dir for peers in jald.cfg

   ```
   sudo keytool -exportcert -rfc -keystore server.jks > server.pem
   cp server.pem ~/JALoP/test-input/certs/
   ```

4. **VERY IMPORTANT!!! You must create a link to the copied server.pem with the format <cert hash>.0 in the remote certs jald dir: ~/JALoP/test-input/certs/ or you will get an “unknown_ca” error on jjnl connect.**

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

5. Create the JALoP subscriber truststore, by importing the publisher.crt from the remote jalop peer, created in step 1 above, Enter “yes” to complete the import when prompted.

   ```
   sudo keytool -importcert -keystore remotes.jks -file <path to publisher.crt> -storepass changeit -keypass changeit -noprompt -alias jalop_publisher
   ```

6. Use the following jald.cfg file and update paths if needed for jald

```
# The path to the private key, used for TLS.
private_key = "./test-input/publisher.key";

# The path to the public cert, used for TLS.
public_cert = "./test-input/publisher.crt";

# The directory containing the certificates for the remote peers.
remote_cert_dir = "./test-input/certs/";

# The path to the root of the database.
```

```
db_root = "./testdb";

# The path to a directory containing the JALoP schemas.
schemas_root = "./schemas";

# The port the Publisher will listen on.
port = 1234L;

# The IP address (interface) the Publisher will to listen on, or 0.0.0.0 to listen on all.
Host = "<enter jald host ip address here>";

# For subscribe, the maximum number of records to send before sending a 'digest' message
pending_digest_max = 10L;

# For subscribe, the maximum number of seconds to wait, before sending a 'digest' message
pending_digest_timeout = 100L;

# How long to wait, in seconds, before polling for records after finding no records
poll_time = 1L;

# List of allowed Subscriber peer configurations
# List of allowed Subscriber peer configurations
peers = ( {
hosts = ("<enter jjnl subscriber ip address here>");
digest_challenge = "on";
subscribe_allow = ("journal", "audit", "log");
cert_dir = "./test-input/certs/";

        } );
```

7. Start jald

```
cd ~/JALoP
      jald –no-daemon -c <path to config file above>
```

8. Use the sampleSubscriber.json config file for jjnl subscriber:

```
{
  "address": "<ip address of jald publisher to connect>",
  "port": 1234,
  "subscriber": {
        "sessionTimeout": "00:20:00",
        "dataClass": [ "audit", "log", "journal" ],
        "pendingDigestMax": 1,
        "pendingDigestTimeout": 120,
        "output": "./output",
        "mode": "archive",
  }
  "ssl": {
    "Key Algorithm": "SunX509",
    "Key Store Passphrase": "changeit",
    "Key Store Data Type": "file",
    "Key Store": "keystore/server.jks",

    "Trust Algorithm": "SunX509",
    "Trust Store Passphrase": "changeit",
    "Trust Store Data Type": "file",
    "Trust Store": "keystore/remotes.jks",
  }
}
```

9. Start the jjnl subscriber to connect to jald using config file above

```
cd ~/jjnl/jnl_test/target
      java -jar jnl_test1.0.0.jar <path to config file above>
```

10. TLS connection should be successful.