**Configuring TLS Mutual Authentication for TKW and Supplier’s Systems (including 111)**

**Strategy**

TLS Mutual Authentication requires that all aspects are correct and present. The nature of security is that unless this is true then authentication fails and no messages are sent/received. It is therefore not always clear why mutual authentication is not working so we must try to eliminate as many of the variables as possible. The best approach therefore is to demonstrate that the test tool (TKW) works first, then prove that the system under test works with this verified test tool. The best way to verify the test tool is to install 2 instances of it and verifiably transmit/receive messages using TLS MA.

**TKW-TKW Configuration**

Start with 2 clean installations of TKW using the latest version of the installer. These 2 installations may or may not be on the same physical machine. Install them into the folders:

TKW\_HOST

TKW\_CLIENT

… including the relevant ‘config’ directory, the relevant test messages directory and the test certificates directory.

**Trusted Certificates – Installing the root CA and sub CA certificates**

Using the rootCA and subCA found in \TKW\contrib\Test\_Certificates\TLS\_Test\_Certificates[[1]](#footnote-1)\* and the Java keytool (which is part of the library of resources installed with Java) import both these trusted certificates into the local JVM. This will have to be repeated on any subsequent machine if 2 physical machines are being used.

*keytool -import -trustcacerts -file TKWCA.crt[[2]](#footnote-2)\* -alias ROOTCA\_ALIAS -keystore $JAVA\_HOME/jre/lib/security/cacerts*

*keytool -import -trustcacerts -file tkwtlssubca.crt[[3]](#footnote-3)\* -alias SUBROOTCA\_ALIAS -keystore $JAVA\_HOME/jre/lib/security/cacerts*

where “$JAVA\_HOME/jre” is the directory of the latest version of JRE held locally. The default keystore password is changeit

Verify that these certificates have been successfully imported using

*keytool -list -v -keystore cacerts*

Note: If there have been any versions of these certificates installed previously into the Java JVM cacerts store it may be preferable to remove these. A list can be displayed using:

*keytool -list -v -keystore $JAVA\_HOME/jre/lib/security/cacerts*

and certs can be deleted using:

*keytool -delete -alias mydomain -keystore keystore.jks*

Setting up the test certificate directory/Keystore

Copy the test certificates into the certs directory:

*Copy \TKW\_CLIENT\TKW\contrib\Test\_Certificates\Test01\test01.jks\* to TKW\_CLIENT\TKW\config\[config]\certs*

*Copy \TKW\_HOST\TKW\contrib\Test\_Certificates\Test02\test02.jks\* to TKW\_HOST\TKW\config\[config]\certs*

**Configuring the CLIENT properties file**

In addition to the usual changed that might be made to the tkw.properties file, edit the \TKW\_CLIENT\TKW\config\[config]\tkw.properties file. Initially use TEST01.jks (or the 111 equivalent) and ensure that the following lines are updated to:

#

Hashed out truststore and password

tks.receivetls Yes

tks.sendtls Yes

#

…

#

#tks.tls.truststore C:/TKW\_CLIENT/TKW/config/[config]/certs/tls.jks

#tks.tls.trustpassword password

tks.tls.keystore C:/TKW\_client/TKW/config/[config]/certs/test01.jks

tks.tls.keystorepassword test01tls\_moscow

This password value is taken from password.txt file in the Test Certificate directory

#

…

#

tks.tls.servermutualauthentication No

Subject DN test used for 111 systems HOST ONLY so is hashed out

#tks.tls.ma.filterclientsubjectdn oneoneone.nhs.uk

tks.tls.clientmutualauthentication Yes

#

…

#

tks.transmitter.send.url <https://127.0.0.1:443/syncsoap>

#

Secure Port 443 is used by convention

**Configuring the HOST properties file**

In addition to the usual changed that might be made to the tkw.properties file, edit the \TKW\_HOST\TKW\config\[config]\tkw.properties file. Initially use TEST02.jks (or the 111 equivalent) and ensure that the following lines are updated to:

#

Hashed out truststore and password

tks.receivetls Yes

tks.sendtls Yes

#

…

#

#tks.tls.truststore C:/TKW\_CLIENT/TKW/config/[config]/certs/tls.jks

#tks.tls.trustpassword password

tks.tls.keystore C:/TKW\_client/TKW/config/[config]/certs/test02.jks

tks.tls.keystorepassword test02tls\_berlin

This password value is taken from password.txt file in the Test Certificate directory

#

…

#

tks.tls.servermutualauthentication Yes

Subject DN test used for 111 systems HOST ONLY, so may or may not be hashed out

tks.tls.ma.filterclientsubjectdn oneoneone.nhs.uk

tks.tls.clientmutualauthentication No

#

…

#

#

Secure Port 443 is used by convention

tks.Toolkit.listenaddr localhost

tks.Toolkit.listenport 443

#

**Executing TKW-TKW Mutual Authentication**

Taka a message from the appropriate Test Message directory and copy it into the CLIENT configuration transmitter\_source directory.

Start the HOST in simulator mode:

*java -jar TKW.jar -simulator tkw.properties*

Transmit the message from CLIENT in transmit mode:

*java -jar TKW.jar -transmit tkw.properties*

**Verifying that Mutual Authentication has been successful**

**TKW\_CLIENT**:

Log file:

The following lines indicate that the truststore has not been set in the tkw.properties and are not significant as we are using JVM cacerts:

*Location: Location not given : Message: Warning: Property tks.tls.truststore not set explicitly, if not set in JVM cacerts, TLS initialisation may fail*

*Location: Location not given : Message: Warning: Property tks.tls.trustpassword not set explicitly, if not set in JVM cacerts, TLS initialisation may fail*

These lines indicate that MA has been successful:

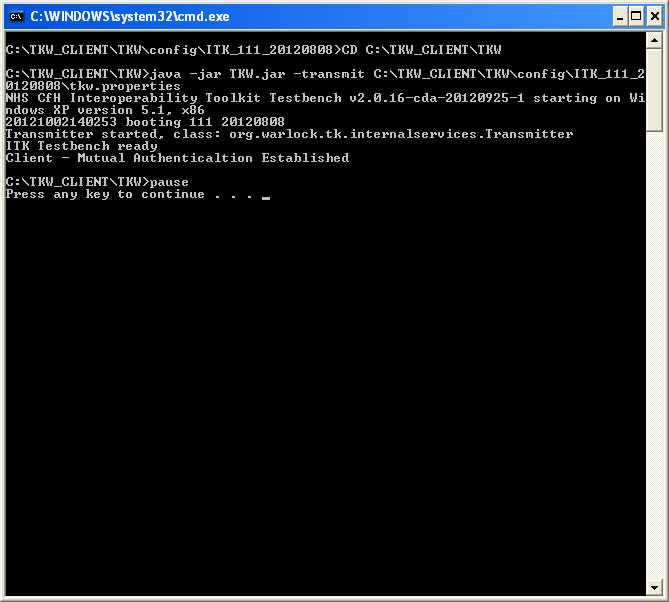
*Location: Sender : Message: Sending to https://127.0.0.1:4848/syncsoap*

*Location: Location not given : Message: Client - Mutual Authentication Established*

transmitter\_sent\_messages folder:

This folder will contain a .log file with the original transmitted file and the response from TKW\_HOST appended at the bottom and an associated .log.signature file.

Command execution prompt:



**TKW\_HOST:**

Log file:

The following lines indicate that the truststore has not been set in the tkw.properties and are not significant as we are using JVM cacerts:

*Location: Location not given : Message: Warning: Property tks.tls.truststore not set explicitly, if not set in JVM cacerts, TLS initialisation may fail*

*Location: Location not given : Message: Warning: Property tks.tls.trustpassword not set explicitly, if not set in JVM cacerts, TLS initialisation may fail*

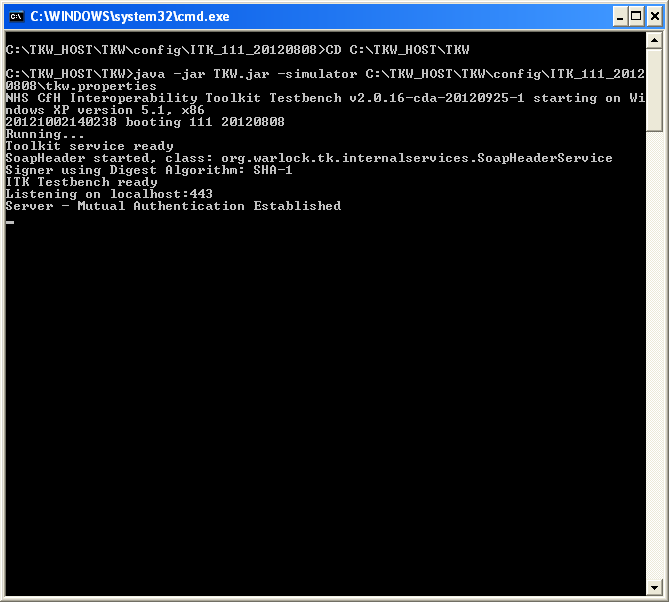
These lines indicate that MA has been successful:

*Location: Sender Location not given : Message: Server - Mutual Authentication Established*

simulator\_saved\_messages folder:

This folder will contain a .log file with the original inbound message from TKW\_HOST and the response appended at the bottom and an associated .log.signature file.

Command execution prompt:



**Next Steps**

Now that mutually Authenticated messaging has verifiably taken place between 2 instances of TKW, substitute the supplier’s system for the appropriate instance of TKW. Remember to set up the identical trust certs and test certs/keystores within the supplier system as were in the associated TKW instance.

Now if there are any issues sending/receiving messages using TKW and the supplier system it will be clear that the former is in error.

After successful TLS MA between TKW and the supplier’s system then proceed to testing the system using the different test certificates as directed by the ITK accreditation team. These test certs/keystores can also be tested using the TKW-TKW model.

**Troubleshooting:**

***The simulator does not state that it is listening on host:port (e.g. localhost:443).*** This indicates that there is something wrong with the TKW\_HOST keystore/test java key store. Often this is an incorrect reference to the .jks location or incorrect password. Remember that java keystore files (.jks) are required here not certificates .crt

***The transmitter seems to have sent ok but nothing in HOST simulator\_saved\_messages. There also is no .log.signature file in the transmitter\_sent\_messages directory***

This indicates that there is something wrong with the TKW\_CLIENT keystore/test java key store. Often this is an incorrect reference to the .jks location or incorrect password. Remember that java keystore files (.jks) are required here not certificates .crt

***TKW\_CLIENT command line responds with “Failed to send message to address https://127.0.0.1:443/syncsoap : sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target : Flushing header”***

***TKW\_HOST command line responds with “Server - Mutual Authentication Failed Exception checking peer subject DN: javax.net.ssl.SSLPeerUnverifiedException: peer not authenticated”***

This indicates that the truststore is incorrect – check that the correct versions of the root CA and sub CA .crt files have been successfully imported into cacerts

1. \* [↑](#footnote-ref-1)
2. \* or 111 equivalent if necessary [↑](#footnote-ref-2)
3. \* [↑](#footnote-ref-3)