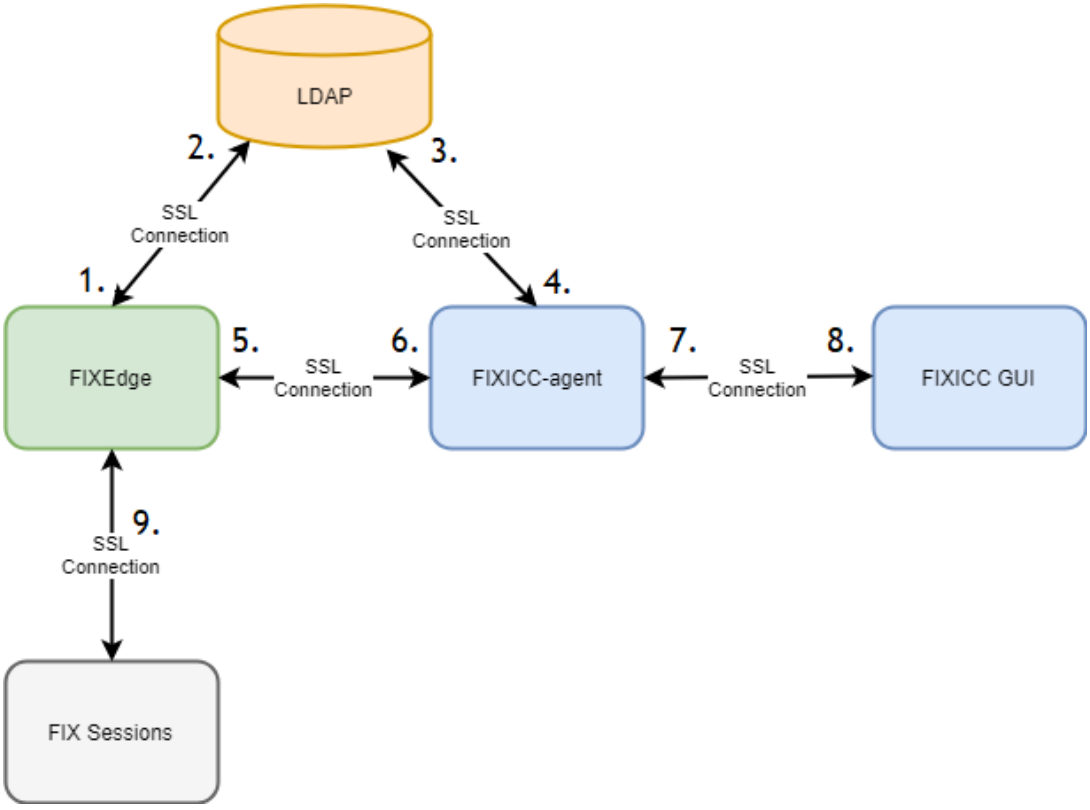
How to setup secure SSL/TLS conntion between FIXEdge, FIXICC-agent and FIXICC

Overview

This article shows how to configure a secure connection between FIXEdge's parties as on a diagram below:



1. FIXEdge side. Establish SSL connection to LDAP service (out of scope)
2. LDAP Service side. Accept connection from FIXEdge (out of scope)
3. LDAP Service side. Accept connection from FIXICC-agent (out of scope)
4. FIXICC-agent side. Establish SSL connection to LDAP service
5. FIXEdge side. Accept SSL connection from FIXICC-agent
6. FIXICC-agent side. Establish SSL connection to FIXEdge
7. FIXICC-agent side. Accept SSL connections.
8. FIXICC GUI side. Establish SSL connections.

9. Enable SSL connections between FIXEdge and FIX-clients

FIXEdge uses OpenSSL for a secure connection.

FIXICC and FIXICC-agent package includes and runs on JRE 1.8.x. Java Virtual Machine determines which secure layer to use, in Java 8 it is TLS 1.2 by default.

This article doesn't describe how to configure SSL connection on LDAP and FIX Sessions sides.

Enabling SSL/TLS connection in java applications (FIXICC GUI, FIXICC-agent)

In order to establish SSL/TLS connection between two java applications, one can use KeyStores and TrustStores:

- KeyStore is used for storing of private keys and certificates. It's commonly used on the server-side.
- TrustStore is used for storing trusted certificates and public keys for trusted certificate authorities CA and self-signed certificates. It's commonly used on the client-side.

keytool is a official java tool for keys and certificates management. Here is an example of how to create these storages by using keytool that is part of J2SE SDK(http://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html).

Manage KeyStores and TrustStores

The following steps allow creating necessary containers for certificates and keys which would be used for establishing SSL Connections between applications.

Create KeyStore with the certificate

Create a new KeyStore containing a certificate and a private key.

```
keytool -keystore fixiccKeystore.key -genkey -alias fixicc
```

The program will ask for certificate owner information and request to enter a password for the KeyStore.
fixiccKeystore.key - is a resulted KeyStore file.

The further commands and configuration contain <keystore_password> as a placeholder for the password.

Export certificate from KeyStore

Export the public certificate using keytool.

```
keytool -export -keystore fixiccKeystore.key -alias fixicc -rfc -file fixicc.cer
```

The program will request a password *<keystore_password>* for fixiccKeystore.key which was entered during "Create KeyStore with the certificate" step.

Import certificate to TrustStore

Import public certificate to TrustStore

```
keytool -import -file fixicc.cer -alias fixiccUI -keystore fixiccTrustStore.key
```

The program will ask for a new password for Trust store and if the certificate is trusted.

The further commands and configuration contain <truststore_password> as a placeholder for the password.

SSL/TLS Connection Configuration

Enable SSL connections between FIXICC-agent and FIXICC GUI

FIXICC-agent side. Accept SSL connections.

Enable SSL connections in agent.properties:

agent.properties

```
AgentServerEnableSSL=true
```

Use private key and certificate from KeyStore. Pass additional JVM parameters as wrapper parameters in wrapper.conf:

wrapper.conf

```
wrapper.java.additional.1=-Djavax.net.ssl.keyStore=${wrapper_home}/FIXEdge1.fixicc-agent/conf/fixiccKeystore.key
wrapper.java.additional.2=-Djavax.net.ssl.keyStorePassword=<keystore_password>
```

where

${wrapper_home}/FIXEdge1.fixicc-agent/conf/fixiccKeystore.key - the path to keyStore

<*keystore_password*> - the password for keyStore

FIXICC GUI side. Establish SSL connections.

Enable SSL connections in fixengine.properties

fixengine.properties

```
enableSSL=true
```

Use certificate from TrustStore for establishing a connection. Pass additional JVM parameters as 'default_options' parameters in fixicc.conf:

fixicc.conf

```
default_options="<OTHER_PRAMETERS> -J-Djavax.net.ssl.trustStore=etc/fixiccTrustStore.key -J-Djavax.net.ssl.
trustStorePassword=<truststore_password>"
```

where

etc/fixiccTrustStore.key - the path to trustStore

<*truststore_password*> - the password for trustStore

Enable SSL connections between FIXICC-agent and FIXEdge

FIXICC-agent side. Establish SSL connection to FIXEdge

Enable SSL initiator connections in fixengine.properties:

fixengine.properties

```
enableSSL=true
```

Set remote port parameter name in fixicc-agent to SSL port configured in FIXEdge that should be used for SSL connection (i.e. **ListenSSLPort** from engine .properties)

agent.properties

```
EngineProperty.AdminSessionPort = ListenSSLPort
```

Import FIXEdge public certificate (see SSLCertificate parameter value from engine.properties) to TrustStore

```
keytool -import -file fixedge.crt -alias fixedgeSrv -keystore fixiccTrustStore.key
```

Use certificate from TrustStore for establishing a connection. Pass additional JVM parameters as wrapper parameters in wrapper.conf

wrapper.conf

```
wrapper.java.additional.3=-Djavax.net.ssl.trustStore=${wrapper_home}/FIXEdge1.fixicc-agent/conf
/fixiccTrustStore.key
wrapper.java.additional.4=-Djavax.net.ssl.trustStorePassword=<truststore_password>
```

where

${wrapper_home}/FIXEdge1.fixicc-agent/conf/fixiccTrustStore.key - the path to trustStore

<*truststore_password*> - the password for trustStore

FIXEdge side. Accept SSL connection from FIXICC-agent

Enable SSL connections in engine.properties:

engine.properties

```
ListenSSLPort = 8905
SSLCertificate = FIXEdge1/conf/cert.pem
SSLPrivateKey = FIXEdge1/conf/key.pem
SSLProtocols = TLSv1_2
```

where

FIXEdge1/conf/cert.pem - certificate

FIXEdge1/conf/key.pem - private.key

For key and cert files please set required privileges: FIXEdge installation with the principle of least privilege on Linux

For details please refer to:

- How to configure built-in SSL support for FIX session in FIXEdge
- How to use SSL with FIX Antenna C++ and FIX Antenna .NET

Enable SSL connections between FIXICC-agent and LDAP service

FIXICC-agent side. Establish SSL connection to LDAP service

Enable SSL initiator connections in fixengine.properties:

fixengine.properties

```
enableSSL=true
```

Import LDAP server public certificate to TrustStore

```
keytool -import -file ldap.crt -alias ldapSrv -keystore fixiccTrustStore.key
```

Use certificate from TrustStore for establishing a connection. Pass additional JVM parameters as wrapper parameters in wrapper.conf

wrapper.conf

```
wrapper.java.additional.5=-Djavax.net.ssl.trustStore=${wrapper_home}/FIXEdge1.fixicc-agent/conf
/fixiccTrustStore.key
wrapper.java.additional.6=-Djavax.net.ssl.trustStorePassword=<truststore_password>
```

where

${wrapper_home}/FIXEdge1.fixicc-agent/conf/fixiccTrustStore.key - the path to trustStore

<truststore_password> - the password for trustStore

LDAP Service side. Accept connection from FIXICC-agent

Out of the scope of this article.

Enable SSL connections between FIXEdge and FIX-clients

Information can be found in the articles:

- How to configure built-in SSL support for FIX session in FIXEdge
- How to use SSL with FIX Antenna C++ and FIX Antenna .NET

In case if FIX Client doesn't have SSL support in the applications, proxies application like STunnel can be used for it.

See an example of the configuration here: How to configure stunnel to enable SSL for FIX session#InitiatorFIXsession

Enable SSL connections between FIXEdge and LDAP service

Out of the scope of this article.

Troubleshooting

FIXICC-agent. SSL debugging

wrapper.conf

```
wrapper.java.additional.5=-Djavax.net.debug=ssl
```

After setting up configuration you need restart FIXICC Agent. Please check the log file and make sure that the FIXICC Agent started without errors.

Unknown error

To get the reason of unknown errors like below:

```
error:1408A0C1:lib(20):func(138):reason(193). Unknown error 336109761. (Error code = 336109761)
```

run openssl application for error number 1408A0C1

```
openssl errstr 1408A0C1
```

or check the error code meaning reason(193). e.g.: on site

FIXEdge rejects SSL connection from FIXICC-agent

The administrative session from FIXICC-agent is rejected when it is trying to connect over SSL to SSL port with reason:

FixEdge.log

```
Administrative client is rejected: connect port of the client (8905) is different from expected (8900).
```

where:

- 8905 - port accepting SSL connection. See ListenSSLPort parameter.
- 8900 - target port for FIXICC-agent administration session defined in EngineProperty.AdminSessionPort parametert in agent.properties. See FIXICC-agent side. Establish SSL connection to FIXEdge for details.

It can be solved with removing or commenting the propery Monitoring.ListenPort in engine.properties.

Accepting connections of administrative sessions to port from Monitoring.ListenPort stops to work. Non SSL admistrative sessions can be connected to ports from ListenPort parameter in engine.properties.

An example of error output in FIXEdge or FIX Antenna logs:

FixEdge.log

```
<time> Severity=INFO  Category=Engine Incoming TCP connection was detected (from 127.0.0.1:56896).
<time> Severity=WARN  Category=Engine Session <FIXADMIN, AdminClient> : Error during processing Logon message
from 127.0.0.1:56896: Administrative client is rejected: connect port of the client (8905) is different from
expected (8900).
<time> Severity=INFO  Category=Engine Incoming TCP connection was closed (from 127.0.0.1:56896).
```

An example of error output in FIXICC-agent logs:

```
[time] [Thread] DEBUG com.epam.fixicc.transport.fix.FIXTransport - Can't connect to localhost:8905
com.epam.fixicc.transport.TimeOutException: Connection was not established during timeout in 5 sec
        at com.epam.fixicc.transport.fix.FIXTransport.tryToConnect(FIXTransport.java:202) ~[transport-impl-
2.10.21.jar:?]
        at com.epam.fixicc.transport.fix.FIXTransport.connect(FIXTransport.java:155) [transport-impl-2.10.21.
jar:?]
        at com.epam.fixicc.transport.fix.FIXTransport.connect(FIXTransport.java:112) [transport-impl-2.10.21.
jar:?]
        at com.epam.fixicc.agent.engine.transport.EnginePIDStatusListener$ConnectThread.connectInternal
(EnginePIDStatusListener.java:254) [agent-2.10.21.jar:2.10.21]
        at com.epam.fixicc.agent.engine.transport.EnginePIDStatusListener$ConnectThread.run
(EnginePIDStatusListener.java:202) [agent-2.10.21.jar:2.10.21]
[time] [Thread] ERROR com.epam.fixicc.agent.engine.transport.EnginePIDStatusListener - Connection was not
established during timeout in 5 sec
com.epam.fixicc.transport.TimeOutException: Connection was not established during timeout in 5 sec
        at com.epam.fixicc.transport.fix.FIXTransport.tryToConnect(FIXTransport.java:202) ~[transport-impl-
2.10.21.jar:?]
        at com.epam.fixicc.transport.fix.FIXTransport.connect(FIXTransport.java:155) ~[transport-impl-2.10.21.
jar:?]
        at com.epam.fixicc.transport.fix.FIXTransport.connect(FIXTransport.java:112) ~[transport-impl-2.10.21.
jar:?]
        at com.epam.fixicc.agent.engine.transport.EnginePIDStatusListener$ConnectThread.connectInternal
(EnginePIDStatusListener.java:254) [agent-2.10.21.jar:2.10.21]
        at com.epam.fixicc.agent.engine.transport.EnginePIDStatusListener$ConnectThread.run
(EnginePIDStatusListener.java:202) [agent-2.10.21.jar:2.10.21]
```