
Article

[Eduard Lebedyuk](#) · Feb 10, 2023 6m read

Encrypted JDBC connection between Tableau Desktop and InterSystems IRIS

In this article, we will establish an encrypted JDBC connection between Tableau Desktop and InterSystems IRIS database using a JDBC driver.

While [documentation on configuring TLS with Java clients](#) covers all possible topics on establishing an encrypted JDBC connection, configuring it with Tableau might be a little bit tricky, so I decided to write it down.

Securing SuperServer

Before we start with client connections, you need to configure SuperServer, which by default runs on port 1972 and is responsible for xDBC traffic to accept encrypted connections. This topic is described in the [documentation](#), but to summarise it, you need to do three things:

1. Generate CA and server cert/key. I use [easyRSA](#):

```
mkdir /tls
cd /easyrsa3
./easyrsa init-pki
# Prep a vars file https://www.sbarjatiya.com/notes\_wiki/index.php/Easy-rsa#Initialize\_pki\_infrastructure
# cp /tls/vars /opt/install/easy-rsa/easyrsa3/pki/vars
./easyrsa build-ca
./easyrsa gen-req IRIS nopass
./easyrsa sign-req server IRIS
cp pki/issued/* /tls/
cp pki/private/* /tls/
cp pki/ca.crt /tls/
sudo chown irisusr:irissys /tls/*
sudo chmod 440 /tls/*
```

Optionally, generate client cert/key for mutual verification. I recommend doing it after establishing an initial encrypted connection.

2. Create %SuperServer SSL Configuration, which uses server cert/key from (1).

```
set p("CertificateFile")="/tls/IRIS.crt"
set p("Ciphersuites")="TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256"
set p("Description")="Autogenerated SuperServer Configuration"
set p("Enabled")=1
set p("PrivateKeyFile")="/tls/IRIS.key"
set p("PrivateKeyPassword")=""
set p("PrivateKeyType")=2
set p("TLSMaxVersion")=32
set p("TLSMinVersion")=16 // Set TLSMinVersion to 32 to stick with TLSv1.3
set p("Type")=1
```

```
set sc = ##class(Security.SSLConfigs).Create("%SuperServer", .p)
kill p
```

3. Enable (or Require) SuperServer SSL/TLS support::

```
set p("SSLSuperServer")=1
set sc = ##class(Security.System).Modify("SYSTEM", .p)
kill p
```

Where: 0 - disabled, 1 - enabled, 2 - required.

Before you Require SSL/TLS connections, remember to enable SSL/TLS connections to the SuperServer for [WebGateway](#) and [Studio](#). I recommend first enabling SSL/TLS connections, then verifying that all clients (xDBC, WebGateway, Studio, NativeAPI, etc.) use encrypted SSL/TLS connections and requiring SSL/TLS connections only after that.

Now we are ready to establish an encrypted JDBC connection from Tableau Desktop.

Configuring encrypted JDBC connection from Tableau Desktop

1. [Download JDBC drivers](#).

2. Place the JDBC driver into the correct folder:

- Windows: C:/Program Files/Tableau/Drivers
- Mac: /Library/Tableau/Drivers
- Linux: /opt/tableau/tableaudriver/jdbc

3. Create truststore.jks from the SuperServer CA certificate: `keytool -import -file CA.cer -alias CA -keystore truststore.jks -storepass 123456 -noprompt`

4. Create SSLConfig.properties file:

```
debug = false
logfile = javatls.log
protocol = TLSv1.3
cipherSuites = TLS_AES_256_GCM_SHA384
trustStoreType = JKS
trustStore = truststore.jks
trustStorePassword = 123456
trustStoreRecoveryPassword = 123456
```

5. Place SSLConfig.properties and truststore.jks in a working directory of your java program. Note that Tableau spawns several processes. You need a working directory for the Java process (which is not a root process or one of the QT processes). Here's how to find it on Windows with [Process Explorer](#):

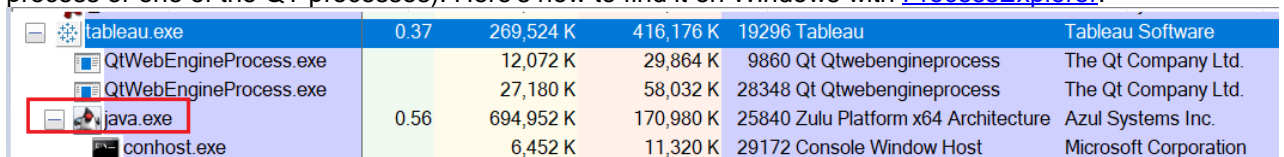


tableau.exe	0.37	269,524 K	416,176 K	19296 Tableau	Tableau Software
QtWebEngineProcess.exe		12,072 K	29,864 K	9860 Qt Qtwebengineprocess	The Qt Company Ltd.
QtWebEngineProcess.exe		27,180 K	58,032 K	28348 Qt Qtwebengineprocess	The Qt Company Ltd.
java.exe	0.56	694,952 K	170,980 K	25840 Zulu Platform x64 Architecture	Azul Systems Inc.
conhost.exe		6,452 K	11,320 K	29172 Console Window Host	Microsoft Corporation

and go to the java process properties and get the current directory:

So for Windows, for me it's C:/Users/lebedyu/AppData/Local/Temp/TableauTemp, or in general %LOCALAPPDATA%/Temp/TableauTemp.

On Mac and Linux, you can figure it out using `ls -ld $(pwd)`. Please write in comments if you determined the directory for Mac/Linux.

6. Create .properties file. This file contains JDBC customizations. Name it IRIS.properties and place it:
 - On Windows: C:\Users<user>\Documents\My Tableau Repository
 - On Mac and Linux: ~/Documents/My Tableau Repository

If you are using the Tableau version before 2020.3, you should use Java 8 for JDBC connections. Java 8 properties files require ISO-8859-1 encoding. As long as your properties file contains only ASCII characters, you can also save it in UTF-8. However, ensure that you don't save with a BOM (Byte order mark). Starting in Tableau version 2020.3, you can use UTF-8 encoded properties files, which are standard with Java 9+.

In this file, specify your JDBC connection properties:

```
host=
port=
user=
password=
connection\ security\ level=10
```

Note that Tableau uses JDBC DriverManager, all properties are [listed here](#). As per the .properties [file spec](#), you must escape whitespaces in keys with a \.

7. Open Tableau and create a new JDBC connection, specifying the properties file. It should connect.

Debugging connections

First try establishing an unencrypted connection. If something does not work, go to My Tableau Repository\Logs and open jprotocolserver.log. In there, search for:

```
Connecting to jdbc:IRIS://host:port/DATABASE
Connection properties {password=*****, connection security level=*****, port=****
***, host=*****, user=_SYSTEM}
Connected using driver {com.intersystems.jdbc.IRISDriver} from isolatedDriver.
```

If logged connection properties do not contain the properties you expect, Tableau is not seeing your IRIS.properties file.

Verifying that connection is encrypted

1. Close everything except for Tableau, which might communicate with IRIS.
2. Open WireShark and start capturing on the correct network interface. Set Capture Filter to: host <iris-ip-address>.
3. In Tableau, try to connect to IRIS, change the schema, or perform any other action requiring server communication.
4. If everything is properly configured, you should see Packets with Protocol TLSv1.3, and following a TCP stream should result in you looking at a ciphertext.

Conclusion

Use the best security practices for xDBC connections.

Links

- [SuperServer TLS](#)
- [WebGateway TLS](#)
- [Studio TLS](#)
- [Java TLS](#)
- [Java Connection Properties](#)
- [easyRSA](#)

[#Encryption](#) [#Java](#) [#JDBC](#) [#SQL](#) [#SSL](#) [#InterSystems IRIS](#)

Source

URL: <https://community.intersystems.com/post/encrypted-jdbc-connection-between-tableau-desktop-and-intersystems-iris>