Managing HTTPS/SSL on server

By default, the Authentication Server runs with HTTPS enabled, using a self-signed certificate that is created on the first service startup. To change HTTPS settings please edit Authentication Server configuration file *./config/authserver.properties* and change related parameters. After the Authentication Server configuration is updated, the service must be restarted. See the HTTPS/SSL parameters description in the section Advance authentication server configuration parameters.

Self-signed server certificate

By default, the Authentication Server uses a self-signed certificate that is created on the first service startup. This means that web browsers will warn users about untrusted server certificate when they first access the Authentication Server. When users choose to trust server certificate, the warning message disappears.

To create the certificate and the keystore, parameters from *<TWCloud directory>/AuthServer/config/authserver.properties* configuration file are used. The keystore will be created automatically on the first server startup if the following conditions are met:

- A keystore file does not already exist in the filesystem.
- All parameters in **authserver.properties**, including the optional ones are set to a non-empty value.
- The keystore type is JKS.

To create a new keystore with other parameters just delete the existing one from the filesystem and restart the service.

CA signed server certificate

For production environments it is highly recommended to use a certificate signed by trusted certificate authorities (CA). The following steps should be done to generate a keystore file providing that you already have a private key and certificate signed by trusted CA.

When executing the **openssl** and **keytool** commands you will be asked for a keystore password. Please read the instructions carefully and provide all required information.

To generate a keystore file

---

1. Create a PKCS 12 file with the OpenSSL tool.

   ```
   openssl pkcs12 -export -in server.crt -inkey server.key -certfile server.crt -out server.p12
   ```

2. Run with administrator rights to create the JKS keystore (Keytool utility can be found in the JRE or JDK bin directory, for example, *C:\Program Files\Java\jre1.8.0_144\bin\keytool.exe*):

   ```
   keytool -importkeystore -srckeystore server.p12 -srcstoretype pkcs12 -destkeystore keystore.jks -
   deststoretype JKS
   ```

3. Copy the file **keystore.jks** to the *./config* directory of the Authentication Server.

> **Note for Windows users**
>
> - You can download OpenSSL binaries for Windows operating system from http://gnuwin32.sourceforge.net/packages/openssl.htm.
> - All commands should be run with administrator rights in the directory where the openssl executable resides.

Deployment on cluster

If the Authentication Server is deployed on a cluster, all service instances should use the same keystore. When using an automatically created keystore with a self-signed certificate, just copy the keystore file from one instance to all the other ones.