Configure TLS for IAM Console

Introduction

The TLS protection of Soffid IAM Console is applied through the configuration of the Apache TomEE embedded in the installation.

This solution is running under java technology therefore we need a jks file (Java Key Store) or a PKCS#12 file with the information of your certificate.

Once you have the Console installed and your certificate in jks format you can follow this steps to configure it the first time or for an update.

Mind that sometimes, the network encryption algorithm is named SSL, in fact, the configuration file still displays the word SSL. However, SSL protocol is now outdated, and TLSv1.2 is used instead.

Load a PKCS#12 (.PFX) file

There are many standard ways to store and transfer private keys and certificates, but the most common one is the PKCS#12 format. Its main advantage is that it contains, in a single file, both the private key and the public certificate.

To transform the .PFX file to a java key store (.JKS), and can use the next command (you have to adapt it to your system):

> Replace <YOUR_FILE> for your file name.

```
keytool -v -importkeystore -srckeystore <YOUR_FILE>.pfx -srcstoretype PKCS12 \
  -destkeystore /opt/soffid/iam-console-4/conf/<YOUR_FILE>.jks \
  -deststoretype JKS \
  -destkeypass 12345678 -srcstorepass 12345678 -deststorepass 12345678
```

Next, you will be asked for the PFX encryption password. It must be provided to you along the PFX file.

Next, you will be asked (probably twice) for the password to be used to encrypt the .JKS file. This password must be written down in the server.xml file. At the sample SSL configuration file placed at the top of this page, the sample password is 123456.

> If we have a key and certificate (key, crt), the step to perform the conversion is:

```
openssl pkcs12 -inkey keyfile.key -in certificate.crt -out name_to_export -export


keytool -v -importkeystore -srckeystore name_to_export -srcstoretype PKCS12
    -destkeystore /opt/soffid/iam-console-4/conf/new_certificate_name.jks
    -deststoretype JKS
    -destkeypass password
    -srcstorepass password
    -deststorepass password
```

Configuration

The configuration file to modify is the following one:

> /opt/soffid/iam-console-4/conf/server.xml

It can contain one or more connectors. Uncomment or add the following one, that enables the TLS configuration:

These are the attributes that you have to configure.

Attribute	Comment
port	You can choose the standard 443 or another custom port
protocols (inside SSLHostConfig tag) **sslEnabledProtocols** (inside Connector tag)	You can configure the protocols allowed. For intance, protocols="TLSv1.3" or sslEnabledProtocols="TLSv1.3"
certificateKeystoreFile	The source by default starts from /opt/soffid/iam-console-4/ (the installation directory)

Attribute	Comment
certificateKeystorePassword	The password used to encrypt the jks file
certificateKeyAlias	The alias to identify your key and certificate

To know the Key Alias, you can run:

```
keytool -list -keystore yourcert.jks
```

Then, copy or replace your jks file into to the file /opt/soffid/iam-console-4/conf/yourcert.jks

After that, you have to restart the iam-console services.

```
sudo systemctl restart soffid-iamconsole
```

> If you have some configuration error, you can search for more information in the Console log (the current day log):
> /opt/soffid/iam-console-4/logs/soffid-YYYY-MM-DD.log

Example server.xml

This example only allows protocols TLSv1.3

```
.............
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
        This connector uses the NIO implementation. The default
        SSLImplementation will depend on the presence of the APR/native
        library and the useOpenSSL attribute of the
        AprLifecycleListener.
        Either JSSE or OpenSSL style configuration may be used regardless of
        the SSLImplementation selected. JSSE style configuration is used below.
    -->

    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true">
        <SSLHostConfig protocols="TLSv1.3">
            <Certificate certificateKeystoreFile="conf/yourcert.jks" certificateKeystorePassword="XXXXXX"
```

```
                certificateKeyAlias="1" type="RSA" xpoweredBy="false" server="Apache TomEE" />
        </SSLHostConfig>
    </Connector>
.............
```

Further information

Additional information can be found at Tomcat website: https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html

---

https://es.wikipedia.org/wiki/TLS

Revision #20
Created 17 March 2021 10:27:12
Updated 10 June 2026 09:00:15 by Sion Vives