Implementing Transport Layer Security (TLS) for SAS Studio

SAS Studio Basic is not configured for HTTPS or encryption between the web application server and the SAS Workspace Server by default. The steps to set-up HTTPS between the web browser and the web application server that hosts SAS Studio are different from the usual TLS/SSL implementation steps for other SAS installations.

The following steps use a self-signed certificate to configure TLS for SAS Studio. You may elect to generate a site signed or a CA signed certificates.

1. Create a new private key and keystore. Save this information in the keystore file named **studio.keystore**.

- **UNIX environments**:

  At a command prompt, enter this command:

  ```
  $SAS_HOME/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/keytool -genkey
    -alias studio -keyalg RSA -keystore
  STUDIO_CONFIG_Directory/appserver/studio/conf/studio.keystore -storepass changeme -keypass changeme -validity
  360 -keysize 2048
  ```

- **Window environments:**

  Open a Windows command prompt as an administrator. Use the `cd` command to your SASHome directory: The default path for SASHome is `C:\Program Files\SASHome`. However, this location might be different at your site.

  ```
  SASHome\SASPrivateJavaRunTimeEnvironment\9.4\jre\bin
  ```

  At the command prompt, enter this command. Be sure to specify the values for KEYSTORE, STOREPASS, and KEYPASS options at your site.

  ```
  keytool -genkey -alias studio -keyalg RSA -keystore C:\sas\studioconfig\apps
  erver\studio\conf\studio.keystore -storepass changeme -keypass changeme    -
  validity 360 -keysize 2048
  ```

Option	Details
KEYSTORE	Specify the full path to the studio.keystore file. By default, the path for the `studioconfig` directory is `C:\sas\studioconfig`.
STOREPASS	Specify the password for the keystore.
KEYPASS	Specify the password for the private key in the studio entry

These passwords must be identical. Keep organizational details handy in case you are prompted for these values. The certificate creation process is dependent on these values and certificates could not be created without them.

2. Change the permissions on the keystore file (studio.keystore) to be readable only by members of the appropriate group. The permissions should be Read/Write only.
3. Open the `STUDIO_CONFIG_Directory/appserver/studio/conf/server.xml` file. Make the highlighted changes to the code for the `Connector` element.

```
<Connector acceptCount="100"
   connectionTimeout="20000"
   executor="tomcatThreadPool"
  keystoreFile="${catalina.base}/conf/studio.keystore"
   keystorePass="password"
   keyAlias="studio"
   maxKeepAliveRequests="15"
   port="38443"
   protocol="org.apache.coyote.http11.Http11Protocol"
redirectPort="38443"
   SSLEnabled="true"
   scheme="https"
   secure="true" />
```

4. Restart the web application server.

- **UNIX environments:** Enter this command at the command prompt:

```
STUDIO_CONFIG_Directory/sasstudio.sh restart webapp
```

- **Windows environments:** Use the Windows Services menu to restart the web application server (*SASStudioWebAppServer, or it could be named tcruntime-c-sas-studioconfig-appserver-studio*).

5. Open SAS Studio using the new URL and the updated port number of 38443. Here is an example:

```
https://machine-name.com:38443/SASStudio
```

Browser

In our example we used a self-signed certificate for the TLS implementation resulting in most browsers displaying a warning when you use this URL. To remove the warning, configure your browser to trust the self-signed certificate for the SAS Studio web

application, or use a certificate from an official certificate authority to configure TLS for SAS studio. Using a certificate authority certificate will mitigate the browser warning.