**SSL configuration for Apache Tomcat & HTTP server**

**SSL configuration for Apache HTTP server in Windows:**

1. Create a new folder in the following path: “**C:\Program Files\Common Files\SSL**”.
2. Copy the **openssl.cnf** file from “**{****Apache24 home dir}\conf\ openssl.cnf**” to “**C:\Program Files\Common Files\SSL**”

(**Note:** Please add “{Apache24 home dir}\bin” under “Path” environment variable if not present)

1. Create a new System Environment Variable:

Variable name: “**OPENSSL\_CONF”**

Variable value : “**{Apache24 home dir}\conf\openssl.cnf”**

1. Get your SSL certificates and save them in the following path “**{Apache24 home dir}\conf\”**. But if you need to generate own certificates, run the below commands in command prompt:
   1. openssl req -config {Apache24 home dir}\conf\openssl.cnf -new -out {Apache24 home dir}\conf\server.csr -keyout {Apache24 home dir}\conf\server.pem

(**Note:** Give “Challenge password” as empty and remember the pass phrase set in the above step.)

* 1. openssl rsa -in {Apache24 home dir}\conf\server.pem -out {Apache24 home dir}\conf\server.key
  2. openssl x509 -req -signkey {Apache24 home dir}\conf\server.key -days 1024 -in {Apache24 home dir}\conf\server.csr -out {Apache24 home dir}\conf\server.crt

1. Edit the file “**httpd.conf”** present in the following path “**{Apache24 home dir}\conf\”** as below:
   1. Uncomment/Edit the below lines:
      1. LoadModule ssl\_module modules/mod\_ssl.so
      2. Include conf/extra/httpd-ssl.conf
      3. LoadModule socache\_shmcb\_module modules/mod\_socache\_shmcb.so
      4. LoadModule rewrite\_module modules/mod\_rewrite.so
   2. Add the below lines at the end of the files
      1. RewriteEngine On
      2. RewriteCond %{HTTPS} off
      3. RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI}
2. Edit “**http-ssl.conf”** file present in “**{Apache24 home dir}\conf\extra\”** folder
   1. ServerName localhost:443
   2. ServerAdmin admin@localhost

**SSL configuration for Apache HTTP server in Linux:**

1. Navigate to “**{Apache24 home dir}/conf”** folder.

We need to install openssl if it is not present. Use the below commands to install openssl

* 1. **apt-get update**
  2. **apt-get install openssl**

1. Onceopenssl is installed use below commands to generate self-signed SSL certificates.
2. openssl genrsa -out server.key 2048
3. openssl req -new -sha256 -key server.key -out server.csr
4. openssl req -new -x509 -nodes -sha1 -days 365 -key server.key -out server.crt
5. Edit the “**httpd.conf”** file present in “**{Apache24 home dir}/conf/”** folder as below:
   1. Uncomment/Edit the below lines:
6. LoadModule ssl\_module modules/mod\_ssl.so
7. Include conf/extra/httpd-ssl.conf
8. LoadModule socache\_shmcb\_module modules/mod\_socache\_shmcb.so
9. rewrite\_module modules/mod\_rewrite.so
   1. Add the below lines at the end of the files:
      * 1. RewriteEngine On
        2. RewriteCond %{HTTPS} off
        3. RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI}
10. Edit “**http-ssl.conf”** file present in “**{Apache24 home dir}/conf/extra/”** folder.
    1. ServerName localhost:443
    2. ServerAdmin admin@localhost
    3. Add the below tag in the file

<VirtualHost \*:80>

ServerName localhost:443

Redirect / https://localhost:443

</VirtualHost>

**SSL configuration for Tomcat (common for Windows and Linux):**

1. Use the following command to generate self signed certificate using keytool.

“keytool -genkeypair -alias Mycert -keyalg RSA -keystore {path\_}\{certificate\_name}.cert”

1. Add the following tag to **server.xml** file present in “**{Apache24 home dir}\conf”**

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" SSLEnabled="true" sslProtocol="TLS" keystoreFile="{path\_to\_certificate\_}" keystorePass="{keyphrase\_}" scheme="https" secure="true" clientAuth="false" />

(NOTE: place this piece of code inside <Service name="Catalina"> tag)

1. Add the following tags to “**web.xml”** file in “**{Apache24 home dir}\conf”**

<security-constraint>

<web-resource-collection>

<web-resource-name>test</web-resource-name>

<url-pattern>/\*</url-pattern>

</web-resource-collection>

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

(NOTE: place this piece of code before </web-app> tag)

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx THE END xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx