**Purpose:**

To install security certificates and modify related jetty files to permit SSL (secure sockets layer), http/2 and https access. Several applications are involved:

* OpenSSL is a library that provides cryptographic functionality to applications such as secure web servers. OpenSSL includes a command line utility that can be used to perform a variety of cryptographic functions. It is described in the openssl manpage. OpenSSL manpages are installed in /usr/local/ssl/man.
* The keytool application manages a keystore (database) of cryptographic keys, X.509 certificate chains, and trusted certificates. Keytool is bundled with the JDK. You can use either Keytool or OpenSSL tools to generate keys and certificates.

Creating secure login involves three steps:

* configure the SSL keystore password and save
* add SSL (secure sockets layer transport) and https modules to start.ini
* modify start.ini for ssl listen port, and update Keystore password

The following process follows that documented by Rishi Khandelwal (2013Sept10) and the Jetty manual.

| **Step** | **Major Activity** | **References, Forms and Details** |
| --- | --- | --- |
| **1** | cd /*usr/*share/jetty9/etc  sudo rm keystore | Move to etc directory  Remove any existing keystore entry |
| **2** | sudo openssl genrsa -des3 -out jetty.key | Generates and responds a key pair in the file jetty.key |
| **3** | sudo openssl req -new -x509 -key jetty.key -days 397 -out jetty.crt | Generates a certificate for the key into the file jetty.crt.  Valid days set to max (397)  Requests information |
| **4** | sudo keytool -keystore keystore -import -alias jetty -file jetty.crt -trustcacerts | Loads a PEM encoded certificate in the jetty.crt file into a JSSE keystore  Will display results, and request acceptance (‘yes’)  Will confirm addition to keystore |
| **5** | sudo openssl req -new -key jetty.key -out jetty.csr | Generates the file jetty.csr using OpenSSL for a key in the file jetty.key  Again asks for information, and a challenge PW |
| **6** | sudo openssl pkcs12 -inkey jetty.key -in jetty.crt -export -out jetty.pkcs12 | Combines the keys in jetty.key and the certificate in the jetty.crt file into the jetty.pkcs12 file.  OpenSSL asks for an export password. A non-empty password is required to make the next step work. |
| **7** | sudo keytool \  -importkeystore \  -srckeystore jetty.pkcs12 \  -srcstoretype PKCS12 \  -destkeystore keystore | Load the resulting PKCS12 file into a JSSE keystore with keytool  Will announce success/failure |
| **8** | cd /usr/share/jetty9/lib  java -cp **jetty-util-9.4.26.v20200117.jar** org.eclipse.jetty.util.security.Password <password> | Obfuscate <password>  NOTE: adjust references to the specific jetty-util jar.  Save the OBF password |
| **9** | cd $JETTY\_BASE  keytool -list \  -keystore etc/keystore \  -storetype jks \  -storepass ''<password>” \  -v | Confirm the keystore contents  Change directory to JETTY\_BASE  Provide in plain format presumed <password>  (quotes optional)  -v for verbose output |
| **10** | sudo java -jar start.jar --add-to-start=ssl | Should respond:  INFO: ssl initialized in  ${jetty.base}/start.ini  INFO: Base directory was modified |
| **11** | Edit start.ini   * Replace first default keystore password (jetty.sslConyext.keyStorePassword) with saved password * set secure listen port (jetty.ssl.port) = 443 * save and exit | Password can be obfuscated or not. This is not encrypted in either case. |
| **12** | Open firewall port 443 permanently   * firewall-cmd --state * firewall-cmd --get-active-zones * firewall-cmd --zone=public --permanent --add-port 443/tcp * firewall-cmd --zone=public --permanent --add-service=https * firewall-cmd --reload * firewall-cmd --list-all * firewall-cmd --list-port | use sudo as necessary  response to –list-port:  8080/tcp 443/tcp  if need to remove ports:   * firewall-cmd --zone=public --permanent --remove-service=https * firewall-cmd --reload |
| 13 | Check listen ports:  netstat -ntlp | grep LISTEN |  |
| 14 | sudo java -jar start.jar --add-to-start=https |  |