KM00756791: **Securing access to ALM Application Server (jetty) using Certificate Authority certificate**

Product Version: ALM 11.5x, 12.x
OS: Windows, Linux

This is published to https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM00756791?lang=en&cc=us&hpappid=202392_OSP_PRO_HPE

This procedure describes configuring secure access to ALM application server (jetty) using Certification Authority (CA) signed certificate. It is a best practice not to use self-signed certificates in production and instead use certificates signed by trusted Certification Authorities.

Basic Steps:

1. Create server certificate in java keystore format.
2. Change jetty configuration to use this keystore.
3. Test secure connection works (you can login to ALM through https protocol).
4. Obfuscate passwords in jetty (recommended).
5. Close non secure connection or redirect http to https.

Step 1: Create server certificate using Certification Authority (CA)

Jetty expects server certificate to be in java keystore (JKS) format.

Option 1: Convert server certificate provided by your Certification Authority

1. Request a server certificate from your Certification Authority. If you need certificate for more than one host and your CA supports Subject Alternative Name (SAN) certificates, you can request one such certificate for multiple hosts using Alternative Names. The alternative names must match how your users are expected to approach these hosts: using short names or long names (FQDN), or both.

2. Export private key with a password that is at least six characters long <mypass>. You will need to provide this password in the following steps. You should now have .pfx file.

3. (optional)View the certificate:
   1. Double-click on .pfx file
   2. Import
   3. Next, Next, Next, Finish
   4. Start->Run->mmc.exe
   5. File->Add/Remove Snap-ins
   6. Certificates
   7. Add
   8. My user account
   9. Finish
   10. OK
   11. Expand Certificates - Current User->Personal->Certificates
   12. Double-click on the certificate and view its details
   13. Certificate can be removed from here after viewing as jetty does not use Windows certificate store.

4. Convert the certificate from PFX/PKCS#12 to JKS format. For example:

   ```
   keytool.exe -importkeystore -srckeystore <mycertificate> -destkeystore <mykeystore> -srcstoretype PKCS12
   ```

   Note: When prompted for password, enter your password <mypass> each time.

5. Import CA root certificate (and any intermediate authority certificates if applicable) into the keystore just created, as in the following example.
   - Download CA certificate in BASE-64 format (<cacert>).
   - Import CA certificate into the keystore:

   ```
   keytool -import -alias <myCA> -file <cacert> -keystore <mykeystore> -storepass <mypass>
   ```

6. Verify that the keystore contains these entries: Trusted Cert Entry and Private Key Entry.

   ```
   keytool -list -keystore <mykeystore>
   ```

   Trusted Cert Entry (root CA)
   Trusted Cert Entry (Intermediate CA)
   Trusted Cert Entry (Issuing CA)
   Private key Entry

You now have server certificate in <mykeystore>. Both the keystore and the private key are password protected with <mypass>.

Option 2: Create a keystore in JKS format manually and have it signed by your Certificate Authority

It is always better to use Certificate Authority UI to create a certificate request using their own templates.
If it is not available, some Certificate Authorities will sign a certificate request file that you produce manually (e.g. using keytool command). Here is how:

1. Generate a keystore with a private key

   ```
   keytool -genkeypair -validity 1065 -keysize 2048 -keyalg rsa -sigalg SHA256withRSA -keystore <mykeystore> -storepass <mypass> -alias <myserver.mydomain>
   ```

   Note: Validity (in days), signature algorithm and keysize depend on your Certificate Authority requirements.
   Note: -sigalg SHA256withRSA is a stronger algorithm and may not be available with some older Certificate Authorities.
   Note: for "What is your first and last name" enter name of the server (short or long/FQDN name -depending on how the users will be approaching your server)

   Note: prior to JDK 1.6 (ALM 11,5x, ALM 12.01), you would use -genkey parameter instead of -genkeypair .

2. Generate a server certificate request to have it signed by your Certificate Authority.

   ```
   keytool -keystore <mykeystore> -storepass <mypass> -alias <myserver.mydomain> -certreq -file CERTREQFILE.csr
   ```

3. Upload this request file or copy/paste its contents into your Certificate Authority interface to be signed by the Authority.
4. Once the certificate has been signed, download the **<signed server certificate>** from your Certificate Authority.
5. Obtain the root authority certificate (and any intermediate authority certificates if applicable).
6. Import the authority certificate(s) obtained in the previous step (root certificate and any intermediate authority certificates if applicable) into the keystore created earlier in this procedure (step 1).

   ```
   keytool -import -trustcacerts -keystore <mykeystore> -storepass <mypass> -alias <rootCA> -file < root CA certificate>
   keytool -import -trustcacerts -keystore <mykeystore> -storepass <mypass> -alias <IntermediateCA> -file < Intermediate CA certificate>
   keytool -import -trustcacerts -keystore <mykeystore> -storepass <mypass> -alias <IssuingCA > -file <Issuing CA certificate>
   ```

   NOTE: if you have not done this for each CA certificate (root and all intermediate), you will see this error during the next step of importing signed server certificate:

   ```
   keytool error: java.lang.Exception: Failed to establish chain from reply
   ```

7. Import the signed certificate into your keystore under the original alias.

   ```
   keytool -import -v -alias <myserver.mydomain> -file <signed server certificate> -keystore <mykeystore> -keypass <mypass> -storepass <mypass>
   ```

   Note: make sure that private key password and keystore password are the same <mypass>.
   You should see: “certificate reply was installed in keystore”. The size of the file will be now significantly larger.

8. Verify that the keystore contains at least two entries: Trusted Cert Entry and Private Key Entry.

   ```
   keytool -list -keystore <mykeystore>
   ```

   Trusted Cert Entry (root CA)
   Trusted Cert Entry (Intermediate CA)
   Trusted Cert Entry (Issuing CA)
   Private key Entry

9. Look at details of the PrivateKeyEntry. Issuer and Owner must now be different. The issuer should be your Certificate Authority (Issuing CA). If you see the same name for the Issuer and Owner, it means your certificate is still self -signed and you need to review the above steps of getting the certificate signed properly.

   ```
   keytool -list –v -alias <myserver.mydomain> -keystore <mykeystore>
   ```

You now have server certificate in <mykeystore>. Both the keystore and the private key are password protected with <mypass>.

Step 2: Change jetty configuration

1. Navigate to the <ALM Deployment Folder>\server\conf directory and make a backup of the jetty-ssl.xml file and the keystore file located in this directory.
2. Copy your keystore file to this directory and rename it keystore.
3. Open the jetty-ssl.xml file, search for "password", and change the value of every password to your password
4. Save jetty-ssl.xml
5. Edit start.ini and uncomment these lines:

   jetty-ssl.xml
   jetty-https.xml (only for ALM 12.20 and higher)

6. Save the start.ini and restart ALM service
7. Check wrapper.log for errors related to keystore. If it works, you should see:

```
INFO   | jvm 1    | 2015/07/01 10:11:12.817 | 2015-07-01 10:11:12.793:INFO:oejs.AbstractConnector:Started SslSelectChannelConnector@0.0.0.0:8443
STARTING
INFO   | jvm 1    | 2015/07/01 10:11:12.817 | Server is ready! (Boot time 80 seconds)
```

Step 3: Test secure connection works
Verify you can login to ALM through https protocol.

Troubleshooting:
If URL does not open even locally:

1. Check on server in wrapper.log for errors.
2. Check on server that listener is running on secure port:
   ```
   netstat -anpo tcp | findstr LISTEN | find <port>
   ```
3. Check from server that SSL handshake is not failing:
   ```
   openssl s_client -connect <server>:<port> -showcerts
   ```
4. If SSL handshake is failing, review the contents of the keystore. It should contain PrivateKeyEntry as described above.
5. Check from server that there are expected protocols (TLS1.2) and ciphers (grade A):
   ```
   nmap --script ssl-enum-ciphers -p <port>  <server>
   ```

If URL opens locally on server, but not remotely from client, check the firewall on server:

secure port should be opened to incoming traffic.

Step 4: Obfuscate passwords in jetty (recommended).

- For ALM/QC versions prior to ALM 12.20:
  - Run this command:
    ```
    <JAVA_HOME>\java \ -cp "<DEPLOYMENT_HOME>\server\lib\*"
    org.eclipse.jetty.http.security.Password <password>
    ```

- For ALM versions starting with ALM 12.20:
  - Determine the version of Jetty that you are using:
    - Locate the < DEPLOYMENT_HOME >\server\lib\jetty-util-<your-jetty-version>.jar file.
    - <your-jetty-version> is the version of Jetty you are using.
  - Run the following commands:
    ```
    $ set JETTY_VERSION=<your-jetty-version>
    <JAVA_HOME>\java -cp <DEPLOYMENT_HOME>\server\lib\jetty-util-
    $JETTY_VERSION.jar org.eclipse.jetty.util.security.Password <password>
    ```
    - For example, run the following command:
      ```
      "C:\Program Files\HP\ALM\ALM\java\jre\bin\java.exe" -cp
      C:\ProgramData\HP\ALM\server\lib\jetty-util-9.1.4.v20140401.jar
      org.eclipse.jetty.util.security.Password changeit
      ```
- Replace the plain text password in the jetty-ssl.xml file with the OBF prefix.
- Restart ALM service

Step 5: Close non secure connection or redirect http to https.

- To close non secure connection, in jetty.xml comment out "addConnector" section with port 8080

- To redirect http to https: see
  https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01733183