MWS

How do I encrypt the connection between MWS and Viewpoint?

Tested on Centos 7.6 using Tomcat 7.0, MWS 9.1.3, Viewpoint 9.1.3, Java 1.8.0

Generate a self signed certificate and private key

First generate a certificate and private key using openssl.

Note: Although it is also possible to generate a certificate using java keytool it is recommended that you use openssl. Exporting a certificate and key from keytool in the PEM format that is required for Viewpoint is much more difficult than it needs to be. The advantage of openssl is that it outputs a PEM formatted certificate and key that is usable in Viewpoint.

First generate a private key

```
==========
   [root]# openssl genrsa -out server.key 2048

==========
```

Create a certificate request and sign it with the private key to create a self signed certificate. The certificate should be valid for 3650 days (~ 10 years)

```
==========
   [root]# openssl req -new -key server.key -out server.csr
   [root]# openssl x509 -req -days 3650 -in server.csr -signkey server.key -out server.crt

==========
```

You should have both a private key (server.key) and a certificate (server.crt). Check that the certificate expiration date is what you expect. By default the certificates expire in 30 days and you may not want to repeat this process each month.

```
==========

   [root]# openssl x509 -in server.crt -text

   ...
      Validity
         Not Before: May  7 16:24:47 2020 GMT
         Not After : May  5 16:24:47 2030 GMT

==========
```

Import the certificate into keytool

You will need to import the certificate into keytool, a utility that comes bundled with Java. This is because MWS runs inside Tomcat and Tomcat applications use keytool to manage certificates and keys for https. By default, keytool saves certificates and keys to a key store that in inside the home directory of the user running it. If you are running Tomcat as the tomcat user (which is the default) you need to switch to the tomcat user so that it gets saved in the .keystore in tomcat's home directory. You may need to adjust /etc/passwd so you can login as the tomcat user (which is disabled by default). You can do this by changing the shell of the tomcat user from /sbin/nologin to /bin/bash. For example:

==========

```
  [root]# vi /etc/passwd
 ...

  tomcat:x:53:53:Apache Tomcat:/usr/share/tomcat:/bin/bash

 [root]# su - tomcat
```

==========

Next use openssl to export the certificate and private key in a pkcs12 encoded file so it can be imported in keytool. Use the name "tomcat" so that Tomcat will find the certificate in keytool.

==========

```
  [tomcat]$ openssl pkcs12 -export -in server.crt -inkey server.key -name tomcat
-out server.pkcs12
 Enter Export Password: changeit
  Verifying - Enter Export Password: changeit
```

==========

Import the certificate and private key into keytool.
Note: Although keytool can import an externally generated PEM encoded certificate it has difficulty importing a PEM encoded private key and attaching it to the certificate. This why it is recommended to convert the certificate and key to PKCS12.
==========

```
  [tomcat]$ keytool -importkeystore -deststorepass changeit -destkeystore
~/.keystore -srckeystore server.pkcs12 -srcstoretype PKCS12
 Enter source keystore password: changeit
  Entry for alias insight-perf02.ac successfully imported.
```

==========
Verify the certificate and key were imported correctly and with an alias of "tomcat"

==========

```
  [tomcat]$ keytool -list
 ...
  Your keystore contains 1 entry

 tomcat, Dec 14, 2018, PrivateKeyEntry,
  Certificate fingerprint (SHA1):
54:30:C0:57:33:17:A9:09:20:5C:0A:E8:56:9C:68:B2:D3:DC:14:90
```

==========

Configure Tomcat to use SSL

Edit the tomcat server.xml file.

==========

```
 [tomcat]$ exit
 [root]# vi /usr/share/tomcat/conf/server.xml
```

==========

Create a Connector element that enables SSL. Note that you use the commented out example of a Connector in server.xml you may need to change the port. In the commented out example the default SSL port for Tomcat is 8443 and that causes a port conflict with Viewpoint File Manager. So in this example the SSL port is changed to 9443. Also the example below assumes Tomcat is running as the tomcat user and that is why the path to the keystore file references the .keystore file in the tomcat user's home directory.

```
 ==========
 <Connector port="9443" protocol="org.apache.coyote.http11.Http11Protocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="${user.home}/.keystore" keystorePass="changeit"
 />
```

==========

It is also advisable to comment out the default unencrypted Connector that runs on port 8080 so that all traffic is forced to use a secure https connection

==========

```
  <!--
  <Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
  -->
```

==========

Start (or restart) tomcat

==========
```
[root]# systemctl restart tomcat
```

==========

Open MWS in a web browser and make sure the changes worked (e.g. https://localhost:9443/mws)

Import the self signed certificate into Viewpoint

Do the following:

==========
```
[root]# cd /opt/viewpoint/lib/requests
[root]# cp cacert.pem cacert.pem.original
[root]# cat /tmp/server.crt  >> cacert.pem
```

==========

Restart Viewpoint

==========
```
[root]# systemctl restart httpd
```

==========

Log into Viewpoint as viewpoint-admin. Got to Basic Configuration and for MWS Configuration > Server put in something like

```
https://localhost:9443
```

Test the configuration.
Unique solution ID: #1223
Author: Nate Seeley
Last update: 2020-05-07 18:27