Adding Listing and Deleting SSL Certificates in keystore Tomcat Application server / What is keystore

Author : admin

I work on ongoing project where *Tomtat Application servers configured to run Clustered* located behind ***Apache*** with **mod_proxy** configured to use *ReverseProxy* are used. One of customers which required a java application deployment experienced issues with application's capability to connect to **SAP database**.

After some investigation I figured out, the *application is unable to connect to the SAP db server* becuse remote host webserver running some SAP related stuff was not connecting due to expired certificate in **Tomcat Keystore** known also as ***JKS / Java Keystore***- (***.keystore***) - which is a **file containing multiple remote hosts imported certificates.**

The best and shortest definition of keystore is:

Keystore entry = private + public key pair = identified by an alias

The keystore protects each private key with its individual password, and also protects the integrity of the entire keystore with a (possibly different) password.

Managing Java imported certificates later used by Tomcat is done with a command line tool part of **JDK** (*Java Development Kit*) called **keystore**. Keystore is usually located under **/opt/java/jdk/bin/keytool**. My Java VM is installed in /opt/ anyways usual location of keytool is **$JAVA_HOME/bin/**

Keytool has capabilities to create / modify / delete or import new SSL certificates and then Java applications can access remote applications which requires Secure Socket Layer handshake . Each certificate kept in **.keystore** file (usually located somewhere under Tomcat web app server directory tree), lets say - **/opt/tomcat/current/conf/.keystore**

1. List current existing imported SSL certificates into Java's Virtual Machine

tomcat-server:~# **/opt/java/jdk/bin/keytool -list -keystore /opt/tomcat/current/conf/.keystore**
password:
Command returns output similar to;

Entry type: trustedCertEntry

Owner: CN=www.yourhost.com, OU=MEMBER OF E.ON GROUP, OU=DEVICES, O=E.GP AG, C=DE
Issuer: CN=E.ON Internal Devices Sub CA V2, OU=CA, O=EGP, C=DE
Serial number: 67460001001c6aa51fd25c0e8320
Valid from: Mon Dec 27 07:05:33 GMT 2010 until: Fri Dec 27 07:05:22 GMT 2013
Certificate fingerprints:
MD5: D1:AA:D5:A9:A3:D2:95:28:F1:79:57:25:D3:6A:16:5E
SHA1: 73:CE:ED:EC:CA:18:E4:E4:2E:AA:25:58:E0:2B:E4:D4:E7:6E:AD:BF
Signature algorithm name: SHA1withRSA
Version: 3

Extensions:

#1: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
DigitalSignature
Key_Encipherment
Key_Agreement
]

#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
CA:false
PathLen: undefined
]

#3: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
[
accessMethod: 1.3.6.1.5.5.7.48.2
accessLocation: URIName: http://yourhost.com/cacerts/egp_internal_devices_sub_ca_v2.crt,
accessMethod: 1.3.6.1.5.5.7.48.2
accessLocation: URIName:
http://www.yourhost1.com/certservices/cacerts/egp_internal_devices_sub_ca_v2.crt]
]

#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D3 52 C7 63 0F 98 BF 6E FE 00 56 5C DF 35 62 22 .R.c...n..V\.5b"

0010: F2 B9 5B 8F ..[.
]
...

Note that password that will be promtped has is by default **changeit** (in case if you don't have explicitly changed it from Tomcat's default config **server.xml).**

2. Delete Old expired SSL host Certificate from Java Keystore
It is good practice to always make backup of old *.keystore* before modifying, so I ran:

tomcat-server:~# **cp -rpf /opt/tomcat/current/conf/.keystore /opt/tomcat/current/conf/.keystore-05-12-2013**

In my case first I had to delete old expired SSL certificate with:

tomcat-server:~# **/opt/java/jdk/bin/keytool -delete -alias "your-hostname" -v -keystore /opt/tomcat/current/conf/.keystore**

Then to check certificate is no longer existent in keystore chain;
tomcat-server:~# **/opt/java/jdk/bin/keytool -list -keystore /opt/tomcat/current/conf/.keystore**

-keystore - option is obligitory it does specify where keystore file is located
-list - does list the certificate
-v - stands for verbose

3. Finally to import new SSL from already expored via a browser url in keystore

tomcat-server:~# **/opt/java/jdk/bin/keytool -importcert -file /tmp/your-hostname.cer -alias your-hostname.com -keystore /opt/tomcat/current/conf/.keystore**

More complete information on how to deal with keystore is available from Apache Tomcat's SSL Howto - a must read documentation for anyone managing Tomcat.