How to Add a New Root Certificate to the Java Key Store

Last Modified on 03/18/2024 6:43 am EDT

Introduction

Assume that the remote exalate node is using a self-signed certificate (or any certficate where the root certificate is not known to the java stack used in the exalate)

Typically you get a PKIX type of error such as:

```
PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certificatio
n path to requested target
```

Add the root certificate by externalizing the cacerts path

Configure docker-compose

Exalate is coming as a docker image and can be deployed using a docker-compose (for instance - Install Exalate for ServiceNow on Docker)

Exalate is a java based application using standard java: the keystore can be found in the

```
# current location of the cacerts path (images with a version < 5.1.0)
# location can change without notice but probably will not as it is pretty standard
#
$JAVA_HOME/jre/lib/security/cacrts
```

To externalize the cacerts file, you can first copy the file out of the container, store it in a configuration folder, and configure the path in the docker-compose.

```
#
# Extract out of docker-compose.yml
#

  snownode:
    restart: unless-stopped
    ports:
      - 9000:9000
    image: idalko/snownode:5.0.19
    depends_on:
      - database #wait for postgres to be started, not for ready
    volumes:
      - ./persist/home:/opt/snownode/data

#
# Add the path to the externalized cacerts
#
      - ./persist/config/cacerts:/usr/lib/jvm/java-8-oracle/jre/lib/security/cacerts
    environment:
# Add your enviroment settings here
      - PGSSLMODE="require"
      - SNOWNODE_PORT=9000
      - SNOWNODE_PG_HOST=database
      - SNOWNODE_PG_DB=snownode
      - SNOWNODE_PG_USER=idalko
      - SNOWNODE_PG_PWD=idalko
    networks:
      - database
      - default
```

Add the certficate to the cacerts keystore

Example of a command for adding a certificate:

```
# It can be that a password is requested - the default is 'changeit'
keytool -importcert -alias "mycertificate" -trustcacerts -keystore cacerts -file ./mycertificate.cer
```

Add the certificate by creating a new docker image

This approach is equal to the externalization and can be used whenever there is no persistent file storage available (such as in some cloud infrastructures)

The approach is to build a new image which contains the root certificate

Create a new docker file and build it:

```
FROM idalko/snownode:5.0.20


CMD mkdir /certs

COPY mycertificate.cer /certs/

CMD keytool -importcert -alias "mycertificate" -trustcacerts -keystore cacerts -file /certs/mycertificate.cer

CMD rmdir -rf /certs
```

ON THIS PAGE