SSL Configuration for RA components using self-signed certificates

Component where SSL can be enabled

Front End (Front end is rendered via NAC)	Back-end Systems
Web UI (Ideally Release operation center i.e. ROC)	NAC -> NES
ASAP UI	NES <--> Agent

Disclaimer: For ASAP UI to run on https pre-requisite is to have Release automation version 4.7.1 build 413.

Pre-requisite (applicable to whole of document)

1. *You will need CA Release Automation components installed*
2. *Java JDK installed which includes several required utilities including jarsigner (gjarsigner does not work). Once installed, you will want to update your path, eg: PATH=$PATH:/usr/java/jdk-<version>/bin*
3. *keytool path should be set to <Base Directory>/jre/bin*

Enabling SSL on Web-UI

Pre-requisite

1. *Base Directory: <Management Server (NAC) Install Directory>/ for e.g.* /usr/local/LISAReleaseAutomationServer

Steps to enable SSL for Web-UI

1. Navigate to base directory, in this case it will be <Management Server (NAC) Install Directory>
2. keytool -genkeypair -keyalg RSA -keysize 2048 -keystore conf/custom-keystore.jks -alias nac-env
3. Modify server.xml on <Management Server (NAC) Install Directory>/conf to contain below values.

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
                    SSLEnabled="true"
                    maxThreads="150"
                    scheme="https"
                    secure="true"
                    clientAuth="false"
                    sslProtocol="TLS"
                    keyAlias="nac-env"
                    keystoreFile="conf/custom-keystore.jks"
                    keystorePass="changeit">
</Connector>
```

Enabling SSL on ASAP GUI

Pre-requisite

1. *Base Directory: <Management Server (NAC) Install Directory>/ for e.g.* /usr/local/LISAReleaseAutomationServer

Steps to enable SSL for ASAP GUI

1. Navigate to base directory, in this case it will be <Management Server (NAC) Install Directory>

2. keytool -exportcert -alias nac-env -file nac.crt -keystore conf/custom-keystore.jks -v –rfc
   Note: custom-keystore.jks is been generated in above section step-2

3. keytool -importcert -alias nac-env -file nac.crt -keystore nolio.jks -v –rfc
   *Note: for ASAP UI the truststore **must** be named nolio.jks*

4. jar cvf custom-truststore.jar nolio.jks
   *Note: for ASAP UI the jar file **must** be named custom-truststore.jar*

5. jarsigner -keystore conf/custom-keystore.jks -verbose -keypass <plaintext password used to create keypair> custom-truststore.jar nac-env

6. cp custom-truststore.jar //webapps/nolio-app/apps/v2.0.0/lib

7. Create file /conf/security-customization.properties and insert ui.trustStorePassword=changeit

8. Restart NAC

Configuring SSL between Management Sever (NAC) and Execution Server (NES)

Pre-requisite

1. *Base Directory: <Management Server(NAC)/Execution Server (NES) Install Directory> for e.g.* /usr/local/LISAReleaseAutomationServer

On Execution Server (NES)

1. Be in base directory in this case it will be <Execution Server (NES) Install Directory>
2. keytool -genkeypair -keyalg RSA -keysize 2048 -keystore conf/custom-keystore.jks -alias nes-env
3. keytool -exportcert -alias nes-env -file nes.crt -keystore conf/custom-keystore.jks -v –rfc
4. keytool -importcert -alias nes-env -file nes.crt -v -rfc -keystore conf/custom-truststore.jks
5. Create or update existing conf/security-customization.properties file and add below entry

   *javax.net.ssl.trustStore=conf/**custom-truststore.jks***
   *javax.net.ssl.trustStorePassword=<plaintext password for **conf/custom-truststore.jks**>*

6. Modify server.xml on <Execution Server (NES) Install Directory>/conf to contain below values.

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
                       SSLEnabled="true"
                       maxThreads="150"
                       scheme="https"
                       secure="true"
                       clientAuth="false"
                       sslProtocol="TLS"
                       keyAlias="nes-env"
                       keystoreFile="conf/custom-keystore.jks"
                       keystorePass="changeit">
</Connector>
```

Note: custom-keystore and custom-truststore were created in step 2 and 4 with password changeit.

On Management Server (NAC)

1. Copy nes.crt created in On Execution Server(NES) section step 3 to Management Server (NAC) tmp directory
2. Navigate to base directory, in this case it will be <Management Server (NAC) Install Directory>
3. keytool -importcert -alias nes-env -file nes.crt -v -rfc -keystore conf/custom-truststore.jks
4. Create file conf/security-customization.properties and add below entry

   *javax.net.ssl.trustStore=conf/**custom-truststore.jks***
   *javax.net.ssl.trustStorePassword=<plaintext password for **conf/custom-truststore.jks**>*

Note: Restart Management Server (NAC) and Execution Server (NES), login to ASAP and add an execution server with port 8443(SSL port) on protocol https

Configuring SSL between Execution Server (NES) and Agents

Pre-requisite

1. *Base Directory: <Agent (NAG)/Execution Server (NES) Install Directory> for e.g.* /usr/local/LISAReleaseAutomationServer

On Agent (NAG)[Section-1]

1. Navigate to base directory in this case it will be <Agent (NAG) Install Directory>
2. keytool -genkeypair -keyalg RSA -keysize 2048 -keystore conf/custom-keystore.jks -alias nag-env
3. keytool -exportcert -alias nag-env -file nag.crt -keystore conf/custom-keystore.jks -v –rfc

On Execution Server (NES)

1. Copy nag.crt generated in section On Agent (NAG) to tmp location on Execution Server (NES)
2. keytool -importcert -alias nag-env -file nag.crt -v -rfc -keystore conf/custom-truststore.jks

   Note: we have already imported nes.crt in custom-truststore.jks in SSL configuration between Management Server (NAC)-Execution Server (NES)

3. Generate encrypted password:
   execute scripts/encrypt_nimi_password.sh <password used for custom-keystore.jks and for custom-truststore.jks>and copy the encrypted password.
   For windows file name will be .bat
   For e.g. if the password for both truststore and keystore is changeit then command to execute is
   $bash: scripts/encrypt_nimi_password.sh changeit
   B5E744BB86EC80C637AE466A33BE5AC4

4. Modify nimi_config.xml on <Execution Server (NES) Install Directory>/conf to contain below values.

```
<security>
   <enabled>true</enabled>
   <keystore>conf/custom-keystore.jks</keystore>
   <keystore_password>B5E744BB86EC80C637AE466A33BE5AC4</keystore_password>
   <trust_store>conf/custom-truststore.jks</trust_store>
   <trustore_password>B5E744BB86EC80C637AE466A33BE5AC4</trustore_password>
</security>
```

On Agent NAG[Section-2]

1. Copy conf/custom-truststore.jks from Execution Server (NES) to Agents conf folder

2. Generate encrypted password:
   execute scripts/encrypt_nimi_password.sh <password used for custom-keystore.jks and for custom-truststore.jks>and copy the encrypted password.
   For windows file name will be .bat

   For e.g. if the password for both truststore and keystore is changeit then command to execute is

   $bash: scripts/encrypt_nimi_password.sh changeit

   B5E744BB86EC80C637AE466A33BE5AC4

3. Modify nimi_config.xml on <Agent (NAG) Install Directory>/conf to contain below values.

```
<security>
   <enabled>true</enabled>
   <keystore>conf/custom-keystore.jks</keystore>
   <keystore_password>B5E744BB86EC80C637AE466A33BE5AC4</keystore_password>
   <trust_store>conf/custom-truststore.jks</trust_store>
   <trustore_password>B5E744BB86EC80C637AE466A33BE5AC4</trustore_password>
</security>
```

Note: Restart Execution Server (NES) and Agent (NAG) services.

Important Notes

- To enable other agents or siblings of the agent configured in section **Configuring SSL between Execution Server (NES) and Agents** to communicate to NES on SSL using same certificate generated above do below
    - Copy custom-truststore.jks from NES (refer Section-2, step-1) to agent conf/ folder
    - Copy custom-keystore.jks from sibling agent which is configured to connect to NES on SSL to current agent conf/ folder (refer Section-1, step-2)
    - Make changes in agent's nimi_config.xml and restart the agent service (refer Section-2, step-3)
- To allow inter-agent communication i.e. agents connected to different execution servers to communicate you need to build trust between these NES's. To enable that follow steps below.

  Note: I will be using NES A and B for explanation purpose, and will try to illustrate the way to build trust between NES A and B so that agents connected to NES A and B can communicate with each other

    - Generate certificate from NES A and NES B executing below command. If you have followed this document you would have already generated nes.crt in step -3 under section **Configuring SSL between Management Sever (NAC) and Execution Server (NES).** If not follow below command to generate certificate
        - keytool -exportcert -alias nes-env -file nes.crt -keystore conf/custom-keystore.jks -v –rfc
    - Assume we generate two certificates nesA.crt and nesB.crt now import these certificate to other NES. i.e. import certificate of NES B into NES A custom-truststore and that of NES A into NES B executing below command on respective NES's

      Pre-requisite

      1. *Base Directory: <Management Server(NAC)/Execution Server (NES) Install Directory> for e.g.* /usr/local/LISAReleaseAutomationServer

      ON NES-A
        - keytool -importcert -alias nesB-env -file nesB.crt -v -rfc -keystore conf/custom-truststore.jks

      ON NES-B
        - keytool -importcert -alias nesA-env -file nesA.crt -v -rfc -keystore conf/custom-truststore.jks

    - Restart NES services.