Virtualize SSL enabled SOAP web services using CA SV

LISABank demo application can be enabled to use SSL for this task, please follow these steps in order to enable SSL on demo server

1 - Please back up files before overwriting it in case if you have to revert it back

2 - Go to DemoServer\lisa-demo-server\jboss\server\default\conf and add/replace following files

client.jks

server.jks

server.truststore

3 - Go to DemoServer\lisa-demo-server\jboss\server\default\deploy\jboss-web.deployer and add/replace

server.xml

4 - Restart demo server and test it

2way ssl = https://localhost:8444/lisabank

1way ssl = https://localhost:8443/lisabank

Plain http = http://localhost:8080/lisabank

After enabling 2-way / mutual authentication in the demo server, you won't be able to access the protected server without the client certificate. To add the client certificate to your web browser, take the "client.jks" file that was created above and convert it into PKCS12 format. NOTE: You need Java 7 and later to do this.

keytool -importkeystore -srckeystore client.jks -srcstoretype JKS -deststoretype PKCS12 -destkeystore client.p12

We are using client.jks file that you used earlier.

Now follow the instructions for your web browser to import the certificate, "client.p12". See this web page for some hints.

http://t2.unl.edu/documentation/gpn/importing-a-user-certificate-in-your-browser

You may get a security exception in your browser while accessing 1 way ssl enabled services as the certs used are self signed, just ignore it and move on.

author@bhavink

5 - Edit your lisa.properties located inside LISA_HOME , this is the root folder of your installation, if you are using a remote registry/vse server than you need to make sure that

ssl certs path is updated appropriately for those systems.

#ex entry for your workstations lisa.prop file - certs are saved localy, you can have certs saved anywhere on the disk

ssl.client.cert.path=C:/itko/lisa/server/9.0GA/DemoServer/lisa-demo-server/jboss/server/default/conf/client.jks

ssl.client.cert.pass=123456

ssl.client.key.pass=123456

ssl.client.alias=clientkey

ssl.server.cert.path=C:/itko/lisa/server/9.0GA/DemoServer/lisa-demo-server/jboss/server/default/conf/server.jks

ssl.server.cert.pass=123456

and restart WS, everytime you edit lisa.properties file you need to restart WORKSTATION and REGISTRY and VSE server

6 - Download demo project and bring it up in WS (WorkStation)

Understand your project

./Configs

contains config or project properties that are used by test cases as well as virtual services

project.config file a DEFAULT config and is available to each and every DevTest asset

One can make a config active by right clicking it and selecting make it active, by doing this every test case of VS that you run will have project + active configuration available to it. This is one of the most important feature of DevTest, this brings us the benefits of properties "inheritance" and "overriding" in DevTest

You'll notice that I have following properties in my project.config

WSSERVER={{WSSERVER}}

WSPORT={{WSPORT}}

ENDPOINT={{ENDPOINT}}

VSENDPOINT={{VSENDPOINT}}

Each key is set to the value of an activated config so if you activate 2wayssl-project.config than project.config will be set to this at runtime

WSSERVER=localhost

WSPORT=8444

ENDPOINT=https\://{{WSSERVER}}\:{{WSPORT}}/itkoExamples/EJB3UserControlBean

VSENDPOINT=https\://{{WSSERVER}}\:7444/itkoExamples/EJB3UserControlBean

The benefit of this approach is that you dont have to edit your test cases or virtual services endpoints in order for them to use relevant port and server's, this is one of the best practices, desiging principles of DevTest, think of each config as an "environment", you create an environment independent DevTest asset and point it to a given environment at runtime, preety cool huh!

./Tests

As the name says, it contains tests and as you can see I have named them appropiately to save you some time/hassle

Let's make sure that we have the right setup

Do not open any test cases, right click each config (1way*, 2way* and plain*) make it active and run Invoke*.tst, this test has two operation, nothing fancy, getUser gets us user details and listUser lists all of the users registered within demobank.

If you have reached here without any problems than you have reached a major milestone in your endeavor of mastering DevTest, you have shown determination to learn something new and an eye for detail, you have earned your first SV badge :) , go ahead pat your back, feel proud about yourself, take deep breath and march on!

7 - Let's record 2way SSL i.e. the client authenticates the server & the server also authenticates the client. (Public cert of client inside server’s truststore and public cert of server inside client’s keystore)

8 - Let's record 1way SSL i.e. the client authenticates the server (public cert of server inside clients keystore)

9 - Deploy it and rerun test case

Now if you are curious about how to create self signed certs and keystores and truststore, try this - you'll need JDK/bin on your path

Create the server keystore

keytool -genkey -alias serverkey -keyalg RSA -keystore server.jks -storepass 123456 -keypass 123456 -dname "CN=localhost, OU=QA, O=iTKO, L=Dallas, ST=Texas, C=US" -validity 3650

Create the server certificate

keytool -export -alias serverkey -keystore server.jks -storepass 123456 -file server.cer

Create the client keystore

keytool -genkey -alias clientkey -keyalg RSA -keystore client.jks -storepass 123456 -keypass 123456 -dname "CN=localhost, OU=QA, O=iTKO, L=Dallas, ST=Texas, C=US" -validity 3650

Create the client certificate

keytool -export -alias clientkey -keystore client.jks -storepass 123456 -file client.cer

Import client certificate into server truststore

keytool -import -v -keystore server.truststore -storepass 123456 -file client.cer

Copy server keystore and server truststore to JBoss / Tomcat conf directory , you can copy it manually also, replace ${LISA_HOME} with absolute path

cp server.jks server.truststore ${LISA_HOME}/DemoServer/lisa-demo-server/jboss/server/default/conf

Credits:

Brad Rogers and Ralph Parker to document “How to tweek our demo server for SSL”

author@bhavink