NanoHUBDeploymentNotes

nanoHUB Deployment Notes

How to install and configure gridshib and SAML tools for gatekeeper

##
Gridshib for GT4
##

1. Install gridshib-gt-source-0_5_1.tar.gz at the gatekeeper

==> download at the http://gridshib.globus.org/download.html

==> ant deploy deploy-echoservice (if previous one is installed, try ant undeploy undeploy-echoservice)

2. Follow the instruction from http://gridshib.globus.org/docs/gridshib-gt-0.5.1/admin-index.html

3. There will be errors if the configurations of "shibechoservice" are not changed.

i) change "GLOBUS_LOCATION" to a real path, e.g. "/opt/globus" at the server-config.wsdd (/opt/globus/etc/gridshib-gt-echo-0_5_1) - you can check by *echo $GLOBUS_LOCATION*

ii) Put the DN at trusted_authn_authorities.txt (/opt/globus/etc/gridshib-gt-echo-0_5_1)-in my case:

/O=Grid/OU=GlobusTest/OU=simpleCA-gatekeeper.rcac.purdue.edu/OU=rcac.purdue.edu/CN=VMware

iii) Start container

4. ### LOG statement settings ###

There is a debug log statement in the SAMLAuthnAssertionPIP module that says "assertion extracted" (with no accompanying error message). In general, better logging statements should be added to both modules (PIP, PDP). One way that exists is to log all SOAP messages to and from the container. You can do this by uncommenting this line in container-log4j.properties:

Enable SOAP Message Logging

log4j.category.org.globus.wsrf.handlers.MessageLoggingHandler=DEBUG

log4j.category.org.globus.wsrf.impl.security.authorization=DEBUG

This will be a little awful but there isn't great logging right now for assertion parsing and query format related work.

Additionally insert following logging statement which is not on the file.

log4j.category.org.globus.gridshib=DEBUG

This logging statement helps a lot!

Then start the container

5. Before running ShibEchoService, Configure in the $GLOBUS_LOCATION/etc/gridshib-gt-echo-0_5_1

1) Change server-config.wsdd

<parameter name="shibecho-SPproviderId" value="https://globus.org/gridshib"/>

==>

<parameter name="shibecho-SPproviderId" value="*urn:mace:inqueue*"/>

<parameter name="shibecho-IdPproviderId" value="https://idp.example.org/shibboleth"/>

==>

<parameter name="shibecho-IdPproviderId" value="_https://shadow120.punch.purdue.edu/shibboleth_"/>

<parameter name="shibecho-AAUrl" value="https://idp.example.org:8443/shibboleth-idp/AA"/>

==>

<parameter name="shibecho-AAUrl" value="_https://shadow120.punch.purdue.edu:8443/shibboleth-idp/AA_"/>

2) Change "echo-attr-authz.xml"* <saml:Attribute AttributeName="urn:mace:dir:attribute-def:countryresident" AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
<saml:AttributeValue>***US***</saml:AttributeValue> : "countryresident" and "US" are added

* <username>user1</username> : "user1" is added

3) Put DN into trusted_authn_authorities.txt (Do not put vmware at the end of the line) as mentioned above

/O=Grid/OU=GlobusTest/OU=simpleCA-gatekeeper.rcac.purdue.edu/OU=rcac.purdue.edu/CN=VMware

4) When using metadata, change "/idp-metadata/metadata.xml"

entityID="https://idp.example.org/shibboleth">

==>

entityID="*urn:mace:inqueue:shadow120.punch.purdue.edu*"

<shibmd:Scope regexp="false">scope.edu</shibmd:Scope>

==>

<shibmd:Scope regexp="false">*purdue.edu*</shibmd:Scope>

Location="https://idp.example.org:8443/shibboleth-idp/AA"

==>

Location="_https://shadow120.punch.purdue.edu:8443/shibboleth-idp/AA_"

* Replace existing attributes with the attribute(with value) to test

<saml:Attribute

xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"

Name="urn:mace:dir:attribute-def:countryresident"

NameFormat="urn:mace:shibboleth:1.0:attributeNamespace:uri">

<saml:AttributeValue>***US***</saml:AttributeValue>

</saml:Attribute>

6. Before running ShibEchoService, put shibboleth-idp CA's cert into /etc/grid-security/certificate

(Add .0 at the file name, e.g. idp-example.crt.0, access mode is rw-r-r-)

Also shibboleth-idp should be able to find gatekeeper's CA cert

This is like:

1) GT has identity G signed by the CA G' (G is typically /etc/grid-security/containercert.pem

2) AA has identity A signed by the CA A' (A is keystore config in Tomcat's server.xml)

3) G queries A

4) G validates A via trusting A' (A' - CA cert - is .0 file in /etc/grid-security/certificates)

5) A validates G via trusting G' (G' - CA cert - is in IdP's meta file, IQ-meta.xml in case of me)

7. Before running ShibEchoService, Configure in the shibboleth-id side (shadow120.punch.purdue.edu)

1) Put gatekeeper's CA cert (simple-CA) to the IQ-meta.xml of IdP from gatekeeper.rcac.purdue.edu

==> scp /etc/grid-security/certificate/ebc****.0, ebc****.signing wlee@shadow120.punch.purdue.edu:/opt/shibboleth-idp/CA_certs/

Then put the key into the IQ-meta.xml file as the other certificate

2) Add nanoHUB ldap cert to java key store file (.jks)

==> sudo keytool -v -alias ldapcacert -import -file nanoldapcert -keystore /opt/cacerts

3) To check the jks file

==> keytool -list -v -keystore /opt/cacerts

4) Change or put "truststoreFile" at the server.xml file of tomcat

==> truststoreFile="/opt/cacerts" truststorePass="*****"

5) Run tomcat again.

8. Run "ShibEchoService" at the client

==> shibecho -s https://gatekeeper.rcac.purdue.edu:8443/wsrf/services/ShibEchoService

##
SAML-TOOLS
##

1. Install gridshib-saml-tools-0_1_3.tar.gz at the client host
==> download at the http://gridshib.globus.org/download.html

2. export GRIDSHIB_HOME=/opt/gridshib-saml-tools-0_1_3

3. Change gridshib-saml-issuer.properties (/opt/gridshib-saml-tools-0_1_3/etc/gridshib/tools/gridshib-saml-issuer.properties)

Format=urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified

formatting.template=%PRINCIPAL%

NameQualifier=urn:mace:inqueue:shadow120.punch.purdue.edu

certLocation=file:///home/wlee/.globus/usercert.pem

keyLocation=file:///home/wlee/.globus/userkey.pem

4. Change mode to 744 for gridshib-saml-issuer, java (/opt/gridshib-saml-tools-0_1_3/)

5. Run

$ gridshib-saml-issuer --user wlee --authn --x509 --outfile /tmp/x509up_u1000 or

$ gridshib-saml-issuer --user user1 --outfile /tmp/x509up_u1000 --authn --authnMethod urn:oasis:names:tc:SAML:1.0:am:password --address 128.210.189.246

6. Checking proxy

$openssl x509 -text -noout -in /tmp/x509up_u1000

then you will see

<assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
AssertionID="_8bf376635a93560be5a3a690437fcb9e" IssueInstant="2007-02-06T18:57:22.984Z" Issuer="O=Grid,OU=GlobusTest,OU=simpleCA-gatekeeper.rcac.purdue.edu,OU=rcac.purdue.edu,CN=VMware" MajorVersion="1" MinorVersion="1"><authenticationStatement AuthenticationInstant="2007-02-06T18:57:19.975Z" AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"><Subject><NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified" NameQualifier="urn:mace:inqueue:shadow120.punch.purdue.edu">user1</NameIdentifier></Subject><SubjectLocality IPAddress="128.210.189.246"></SubjectLocality></AuthenticationStatement><attributeStatement><Subject><NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
NameQualifier="urn:mace:inqueue:shadow120.punch.purdue.edu">user1</NameIdentifier></Subject><attribute AttributeName="urn:oid:1.3.6.1.4.1.5923.1.5.1.1" AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"><attributeValue xsi:type="xsd:string">http://www.nanohub.org</AttributeValue></Attribute></AttributeStatement></Assertion>

##
HOW TO Configure for other services
##

For example of "SecureCounterService" (For gridshib-gt-0.5.1)

1. Change security-config.xml file

==> put <*authz value="counter:org.globus.gridshib.SAMLAuthnPIP counter:org.globus.gridshib.PDP"/>* (counter is random scope)

2. Put parameters at the server-config.wsdd file

==> <parameter name="counter-shibAuthzAttrFile" value="/opt/globus/etc/globus_wsrf_core_samples_counter/attr-authz.xml"/>

<parameter name="counter-AAUrl" value="https://shadow120.punch.purdue.edu:8443/shibboleth-idp/AA"/>

<parameter name="counter-SPproviderId" value="urn:mace:inqueue"/>

<parameter name="counter-IdPproviderId" value="urn:mace:inqueue:shadow120.punch.purdue.edu"/>

<parameter name="counter-trusted-authn-authorities-file" value="/opt/globus/etc/globus_wsrf_core_sample_counter/trusted_authn_authorities.txt"/>

(create trusted_authn_authorities.txt file)

3. Change appropriate attributes at the attr-authz.xml file (e.g. to use countryresident attribute, put one of values)

<saml:Attribute AttributeName="urn:mace:dir:attribute-def:countryresident" AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">

<saml:AttributeValue>***US***</saml:AttributeValue>

4. Run SecureCounterService

==> counter-client -s https://gatekeeper.rcac.purdue.edu:8443/wsrf/services/SecureCounterService