

OSG GUMS/VOMS for WS GRAM Authorization

1. Introduction

1.1 Mapping services

The Globus Toolkit uses mapping services to go from a user's public credential to a local account in the infrastructure. By default, GridMap files are used, which contain a mapping from a user's DN to the local account. GridMap files are mapping tools local to a given resource and are typically installed on the resource itself.

Software services like GUMS and SCAS provide a mapping tool with a remote interface. That is, they provide an interface by which remote clients can contact them to obtain local mapping information for a given user.

1.2 GUMS and XACML Authorization Interface

As a part of the OSG/EGEE Authorization Interoperability effort, [GUMS](#) and SCAS have been augmented with a XACML Authorization service interface. The interface used and the profile these services conform to is described in [OSG/EGEE Authorization Interoperability Profile](#).

In a nutshell, the XACML Authorization interface allows a client to send a set of attributes about the caller, resource and action in the form of an authorization query. The authorization service in turn can use the attributes to determine if the caller is allowed to perform that action on the resource and return an authorization decision. The authorization decision can contain additional obligations.

1.3 OSG Authorization Infrastructure

The OSG infrastructure requires the use of [VOMS](#) and GUMS for the authorization of requests to the services. The following briefly summarizes the steps involved in the authorization of access of an OSG resource:

VOMS server maintains the attributes of the user, for example group membership or VO membership. The user contacts the VOMS server to obtain their attributes and this is embedded in the user's certificate. This information from the certificate, in addition to other certificate information such as DN, is extracted and sent to the GUMS server for authorization. The GUMS server uses all the attributes to determine if the user can access the resource and returns a decision with the local mapping as an obligation.

1.4 Globus Toolkit Callouts

As of GT 4.2.1 release, Globus Toolkit provides a callout that can talk to authorization services that use XACML Authorization Interface and enforce the returned decision. This implies that any deployment of GT Java Web Services can leverage the callouts to talk to an authorization service such as GUMS to determine if a particular service access is allowed. Additionally, policy information points (PIPs) that collect and send attributes as defined in the OSG/EGEE Authorization Interoperability Profile have also been added.

Additionally, [VOMS Authorization Interceptors](#), a Globus Incubator Project provides PIPs that can be plugged into the GT authorization framework and extract the VOMS issued attributes

from a user's certificate. The project also ships an additional PIP that converts the extracted attributes to comply with the OSG/EGEE Interoperability Profile.

The GT authorization framework allows custom plug-ins, deployments that don't comply to the profile, can provide their own PIPs and yet leverage the XACML Authorization Callout to talk to the XACML Authorization Service. The VOMS authorization interceptors can also be used on their own, outside of this context.

Specifically, the following components have been added to Globus Toolkit to enable use of GUMS/VOMS for authorization of WS GRAM service interactions:

- XACML Authorization callout PDP
- PIPs to comply with OSG/EGEE Interoperability profile.
- VOMS PIP (from incubator project)
- Execution service PIPs

1.5 Support outside of OSG infrastructure

As mentioned before, the XACML Authorization Callout and the VOMS PIPs are independent of the OSG infrastructure and can be used in other configurations.

If you have an authorization service that implements the XACML Authorization interface, the XACML Authorization Callout can be configured with the service endpoint and identity and used to contact the service for decisions.

If your authorization infrastructure requires parsing, extraction of VOMS attributes, the the VOMS PIPs can be configured in the security descriptor and used. This can be used in addition to other authorization mechanism to determine if the user has access.

2. Document Scope

This document describes the installation and configuration of the XACML Authorization Callout, in the context of contacting a GUMS service as deployed in the OSG infrastructure, to authorize access to WS GRAM service. Such a setup requires the user to have VOMS attributes embedded in their certificate. This also involves WS GRAM configuration changes to use the required PDPs/PIPs for VOMS attribute parsing and authorization service access.

The document does not describe the installation and configuration of GUMS service. Please refer to [GUMS 1.3](#) documentation for details on setting up and using GUMS.

The document also does not delve into setting up and using of VOMS server. Please refer to [VOMS Documentation](#) for details.

Assumptions:

- A GUMS service endpoint (with XACML Authorization interface) is available
- GUMS service identity is available
- User has contacted a VOMS server and run voms-proxy-init (or equivalent) to obtain VOMS attributes
- GUMS has been set up to trust the VOMS server

3. Installation

The features have been added to GT trunk and 4.2 branch. Download and install GT 4.2.1 release.

- Using GT 4.2.1 Release
 - [GT 4.2.1 Installation](#)
 - In the [Basic Installation](#), use option --enable-gramxacml to install the XACML pieces and the WS GRAM pieces required. Example: ./configure --enable-gramxacml --prefix=\$GLOBUS_LOCATION
 - Follow other installation instructions as described and you will need to install WS GRAM (make wsgram).
- Using make-packages.pl
 - ws co packaging
 - cd packagng
 - export GLOBUS_LOCATION /sandbox/globus/gtInstall
 - ./make-packages.pl --install=\$GLOBUS_LOCATION --packages="globus_java_authz_xacml_test,globus_wsrf_gram_authz_java" - -deps --inplace
 - export GPT_LOCATION /sandbox/globus/gtInstall
 - \$GPT_LOCATION/sbin/gpt-postinstall

The above is for GRAM Fork installation. Refer to GT administrator documentation for enabling other schedulers with GRAM.

The VOMS interceptors for parsing Attribute Certificates from VOMS is not part of the Globus Toolkit yet. It is a Globus Incubator project and needs to be installed separately.

- Installing from source:
 - Set CVSROOT to :pserver:anonymous@cvs.globus.org:/home/globdev/CVS/globus-packages
 - cvs co authz-interceptors/voms
 - cd authz-interceptors/voms/
 - setenv GLOBUS_LOCATION /sandbox/ranantha/gt4.2/gtInstall
 - ant deploy

4. Configuration

4.1 Basic configuration

1. Configure credentials for the GT container as described in [GT Administrator Documentation](#)
2. Configure GRAM as described in [WS GRAM Administrator Documentation](#)

4.2 Container configuration

1. The OSG/EGEE Interoperability profile requires that the host name to be sent as an attribute, rather than IP address. The following parameter in \$GLOBUS_LOCATION/etc/globus_wsrf_core/server-config.wsdd, with in the <globalConfiguration> section:

```
<parameter name="publishHostName" value="true"/>
```

4.3 WS GRAM Factory configuration:

This section configured WS GRAM Factory to use the correct descriptor.

1. WS GRAM factory is configured with security descriptor in file \$GLOBUS_LOCATION/etc/globus_wsrf_gram/server-config.wsdd using property securityDescriptor.
2. A sample descriptor that uses XACML callout is provided and installed at \$GLOBUS_LOCATION/etc/globus_exec_authz_xacml/factory-xacml-voms.xml. This file uses FactoryServicePIP, VOMS AuthzProfilePIP and XACMLAuthorizationCallout to send relevant attributes and request to GUMS/SCAS.
3. Copy the above file as follows:

```
cp $GLOBUS_LOCATION/etc/globus_exec_authz_xacml/factory-xacml-voms.xml
  $GLOBUS_LOCATION/etc/globus_wsrf_gram/factory-xacml-voms-security-
  config.xml
```

4. Modify the above file to provide relevant values for VOMS Authz Profile PIP:
 - VOMS Trust Store: This is configured as parameter "vomsTrustStore" and the value should be set to the directory with certificates for VOMS attributes to be validated.
 - CA Trust Store: This is configured as parameter "caTrustStore" and the value should be set to the directory with trust root information for certificate chain validation.
 - Example:

```
<interceptor name="voms:org.globus.voms.AuthzProfilePIP">
  <parameter>
    <nvparam:nameValueParam>
      <nvparam:parameter name="vomsTrustStore" value="/home/ranantha/
.globus/certificates"/>
      <nvparam:parameter name="caTrustStore" value="/home/ranantha/
.globus/certificates"/>
    </nvparam:nameValueParam>
  </parameter>
</interceptor>
```

- Modify the descriptor to provide relevant values for XACML Authorization Callout
 - Authorization service URL. Example:
<param:authzService url="https://cascade.fnal.gov:8443/gums/services/GUMSXACMLAuthorizationServicePort"/>
 - Authorization service identity. Example:
<param:authzServiceIdentity value="Expected DN of the authz service"/>
- Configure factory service to use the modified security descriptor. Example:
<parameter name="securityDescriptor" value="etc/globus_wsrf_gram/factory-xacml-voms-security-config.xml"/>
- For other optional parameter, refer to 4.5 and 4.6 section.

4.4 WS GRAM Job Resource configuration:

1. WS GRAM resource security descriptor is configured as a JNDI configuration parameter resourceSecurityDescriptorFile in file \$GLOBUS_LOCATION/etc/globus_wsrf_gram/jndi-config.xml.
2. A sample descriptor that uses XACML callout if provided and installed at \$GLOBUS_LOCATION/etc/globus_exec_authz_xacml/job-xacml-voms.xml. The file uses JobServicePIP and XACMLAuthorizationCallout to send relevant attributes and request to GUMS/SCAS.
3. Copy the file as follows:

```
cp $GLOBUS_LOCATION/etc/globus_exec_authz_xacml/job-xacml-voms.xml
  $GLOBUS_LOCATION/etc/globus_wsrf_gram/job-xacml-voms-security-config.xml
```

4. Configure resource security descriptor to point to this file:
 - Edit \$GLOBUS_LOCATION/etc/globus_wsrf_gram/jndi-config.xml to modify resourceSecurityDescriptorFile as follows:
 - <!-- Resource security descriptor -->


```
<parameter>
                <name>resourceSecurityDescriptorFile</name>
                <value>etc/globus_wsrf_gram/job-xacml-voms-security-config.xml</value>
              </parameter>
```
 - Modify the above file to provide relevant values for VOMS Authz Profile PIP:
 - VOMS Trust Store: This is configured as parameter "vomsTrustStore" and the value should be set to the directory with certificates for VOMS attributes to be validated.
 - CA Trust Store: This is configured as parameter "caTrustStore" and the value should be set to the directory with trust root information for certificate chain validation.
 - Example:


```
<interceptor name="voms:org.globus.voms.AuthzProfilePIP">
                <parameter>
                  <nvparam:nameValueParam>
                    <nvparam:parameter name="vomsTrustStore" value="/home/ranantha/.globus/certificates"/>
                    <nvparam:parameter name="caTrustStore" value="/home/ranantha/.globus/certificates"/>
                  </nvparam:nameValueParam>
                </parameter>
              </interceptor>
```
 - Modify the descriptor to provide relevant values for XACML Authorization Callout
 - Authorization service URL. Example:


```
<param:authzService url="https://cascade.fnal.gov:8443/gums/services/GUMSXACMLAuthorizationServicePort"/>
```
 - Authorization service identity. Example:


```
<param:authzServiceIdentity value="Expected DN of the authz service"/>
```
- For other optional parameter, refer to 4.5 and 4.6 section.

4.5 XACML Callout configuration in security descriptor

- XACML Callout PDP configuration is described in detail here [XACML Authz Callout](#)
- OSG Infrastructure requires at least the following:
 - Authorization service identity
 - Authorization service endpoint
 - Privacy enabled

- **Obligation Handlers:**

Custom obligation handlers can be configured as child element of xacmlAuthzParameter element. The handlers should be configured as child element of <param:ObligationHandlers>. Example:

```
<param:ObligationHandlers>
...
</param:ObligationHandlers>
```

Each obligation requires the following configuration element with an obligation id and an obligation handler class.

```
<param:SupportedObligation>
```

```
<param:ObligationId>http://authz-interop.org/xacml/2.0/obligation/
username</param:ObligationId>
```

```
<param:FQClassName>org.globus.wsrfl.impl.security.authorization.LocalAccountObligationHandler</
param:SupportedObligation>
```

4.6 VOMS PIP configuration

The VOMS PIP that complies to the interoperability profile is org.globus.voms.AuthzProfilePIP. It allows for following parameters to be configured:

- VOMS Trust Store: This is configured as parameter "vomsTrustStore" and the value should be set to the directory with certificates for VOMS attributes to be validated.
- CA Trust Store: This is configured as parameter "caTrustStore" and the value should be set to the directory with trust root information for certificate chain validation.

4.7 Sample Security Descriptors

[Sample Factory Security Descriptor \(for GUMS/VOMS\)](#)

[Sample Job Security Descriptor \(for GUMS/VOMS\)](#)

5. Using Clients

1. Use VOMS client software to obtain a proxy with attribute certificates. Link?
2. Set up environment variable to use the proxy in GT clients:
export X509_USER_PROXY /path/to/voms/proxy
3. Use WS GRAM clients [GRAM Client Guide](#)

6. Limitations

1. RFT and Delegation Service have not been modified to use GUMS/VOMS as yet. So WS GRAM cannot be used with file transfer. Refer to [Enabling WS GRAM for OSG/EGEE](#) for details.