Definition and Implementation of a SAML-XACML Profile for Authorization Interoperability across Grid Middleware Implementations in OSG and EGEE

Ian Alderman[9] (alderman@cs.wisc.edu), Mine Altunay[1] (maltunay@fnal.gov), Rachana Ananthakrishnan[8] (ranantha@mcs.anl.gov), Joe Bester[8] (bester@mcs.anl.gov), Keith Chadwick[1] (chadwick@fnal.gov), Vincenzo Ciaschini[7] (vincenzo.ciaschini@cnaf.infn.it), Yuri Demchenko[4] (demch@science.uva.nl), Andrea Ferraro[7] (andrea.ferraro@cnaf.infn.it), Alberto Forti[7] (alberto.forti@cnaf.infn.it), Gabriele Garzoglio[1] (garzoglio@fnal.gov), David Groep[2] (davidg@nikhef.nl), Ted Hesselroth[1] (tdh@fnal.gov), John Hover[3] (jhover@bnl.gov), Oscar Koeroo[2] (okoeroo@nikhef.nl), Chad La Joie[5] (chad.lajoie@switch.ch), Tanya Levshina[1] (tlevshin@fnal.gov), Zach Miller[9] (zmiller@cs.wisc.edu), Jay Packard[3] (jpackard@bnl.gov), Håkon Sagehaug[6] (hakon.sagehaug@bccs.uib.no), Valery Sergeev[1] (vsergeev@fnal.gov), Igor Sfiligoi[1] (sfiligoi@fnal.gov), Neha Sharma[1] (neha@fnal.gov), Frank Siebenlist[8] (franks@mcs.anl.gov), Valerio Venturi[7] (valerio.venturi@cnaf.infn.it), John Weigand[1] (weigand@fnal.gov)

[1] Fermilab, Batavia, IL, USA
[2] NIKHEF, Amsterdam, The Netherlands
[3] BNL, Upton, NY, USA
[4] University of Amsterdam, Amsterdam, The Netherlands
[5] SWITCH, Zürich, Switzerland
[6] BCCS, Bergen, Norway
[7] INFN CNAF, Bologna, Italy
[8] ANL, Argonne, IL, USA
[9] University of Wisconsin, Madison, WI, USA

Keywords: authorization, interoperability, SAML-XACML, OSG, EGEE

Abstract: In order to ensure interoperability between middleware and authorization infrastructures used in the Open Science Grid (OSG) and the Enabling Grids for E-science (EGEE) projects, an Authorization Interoperability activity was initiated in 2006. The interoperability goal was met in two phases: (1) agreeing on a common authorization query interface and protocol with an associated profile that ensures standardized use of attributes and obligations; (2) implementing, testing, and deploying on OSG and EGEE, middleware that supports the interoperability protocol and profile. The activity has involved people from OSG, EGEE, the Globus Toolkit project, and the Condor project. This paper presents a summary of the agreed-upon protocol, profile and the software components involved.

1. Introduction

The Open Science Grid (OSG) [1] and the Enabling Grids for E-sciencE (EGEE) [2] are two major projects devoted to promoting science through the use of distributed, grid computing. Despite the fact that the two projects are mostly independent and operate hardware resources in different parts of the world, a non negligible part of the software stack is shared between the two[1].

Both OSG and EGEE base their authentication infrastructure on Public Key Infrastructure (PKI), leveraging X.509 end-entity and proxy certificates [6,7] for single sign-on and delegation. Initially, both grids based their authorization infrastructures on policies local to resources. With time, however, they extended their infrastructures to centralize the authorization policies at the level of individual sites. In addition, both grids extended their infrastructures to include role-based access to resources, based on a user's Virtual Organization (VO) membership.

While the security model based on a user's VO membership was successfully maintained similar between the two grids, the mechanisms to centralize authorization policies risked diverging. Drawbacks of such divergence consisted in duplication of work and on the requirement that middleware common to both grids supported different authorization plug-ins, depending on the grid on which it was deployed.

[1] Examples of shared software products are the Disk Cache Storage Resource Manager (SRM) [3] and the gLExec identity switching tool [4,5].

The Authorization Interoperability activity was formed in 2006 to address this issue. The collaboration defined a common OSG/EGEE protocol and identity attribute profile for authorization call-out to site-central policy decision services. In lock-step, two independent libraries, one in C and one in Java, have been implemented according to the agreed protocol and profile, and are being used for cross-implementation interoperability.

The activity had resonance with major middleware providers for both grids, namely the Globus Toolkit and the Condor groups. Being active participants to the activity, these groups have started providing middleware that natively supports the common authorization protocol. This greatly simplifies the process of deploying such middleware on both OSG and EGEE.

This paper is organized as follows. Section 2 presents work related to the authorization interoperability activity. Section 3 describes the OSG and EGEE security models. Section 4 summarizes the principal elements of the common authorization interoperability profile. Section 5 discusses how the infrastructures implemented the common profile. Section 6 discusses future work and section 7 presents a summary of the paper.

2. Related Work

The authorization interoperability activity has produced a call-out protocol and identity profile from resource gateways to policy decision services. The activity limited its scope to the EGEE and OSG security model, whereby identities are described via X509 certificates and identity attributes via VOMS [9] attribute certificates. It also targeted a limited set of authorization systems for implementation, namely the Grid User Mapping Service (GUMS) [13] for OSG and the Site Central Authorization Service (SCAS) [27] for EGEE.

The Open Grid Services Architecture (OGSA) Authorization Working Group (WG) of the Open Grid Forum (OGF) [24] is addressing the same problem in a more general context. The objective of the OGSA Authorization WG is to define the specifications needed to allow for interoperability and pluggability of authorization components from multiple authorization domains in the OGSA framework. There are a number of authorization systems emerging in the grid today, in addition to VOMS and XACML (Akenti, Cardea, CAS, PERMIS, etc.); the OGSA-Authorization specifications aim at allowing these solutions to be interchangeably used with middleware that requires authorization functionality. The OGSA-Authorization group leverages authorization work that is ongoing in the Web services world (e.g. SAML, XACML, the WS Security suite) and defines specifications for how these should be used for grid services.

When the Authorization Interoperability activity started, the specification of the OGSA-WG did not address all use cases of interest to the collaboration. Having three members of the OGSA-Authorization WG participating in our activity, we were able to develop a subset of the broader functionality, maintaining consistency with the general direction of the WG.

3. The OSG and EGEE security models

The OSG and EGEE security models are similar in their design. They are both based on PKI, using X.509 end-entity and proxy certificates. Certificates are used to mutually authenticate every request for service. Integrity and confidentiality of the communication is supported both at the transport and message layer, using the standard Transport Layer Security (TLS) protocol [25].

Resources are made available on the grid for user communities, also called Virtual Organizations (VOs). Access to resources is granted to users on the basis of their membership to a VO, rather than on the user's personal attributes.

In the common security model, VOs organize their internal membership structure according to hierarchical groups (e.g. /atlas/usatlas). Members of a group can have special roles for that group (e.g. /atlas/usatlas/Role=SoftwareAdministrator). This structure and relative membership of each user is maintained and published via Virtual Organization Management Servers (VOMS).

Conversely, in the model, resources are grouped according to the administrative boundaries of computing sites. Access to different resources (Storage, Computing, Worker Nodes, etc.) is managed by middleware that acts as a gateway to the resource. To implement access authorization, gateways of both grids obtain the user's membership information and the VO organizational structure from each VO's VOMS. These common sources of attributes, with a well defined access protocol and known unique identities, lay the foundation for interoperations across grids.

Both grids make available to sites lists of member VOs and their preferred privileges at resources; however, sites have the ultimate word on what VOs, VO groups, and VO members are supported and what privileges are granted to them. Typically, privileges are determined by membership in VO groups and roles, like relative priority in a batch system or read/write access to storage areas. Attributes that univocally identify users, like the user's X509 Distinguished Name, are used for some VOs to enable operating system-level protection of concurrently running processes from different users on the same machine.

Users interact with resource gateways on behalf of VOs and VO groups with a certain Role. Before every interaction, the user is responsible for including this information with their credentials. The information is expressed in terms of VO membership attributes, or Fully Qualified Attribute Names (FQANs), and is encapsulated in an Attribute Certificate (AC). The AC is obtained by interacting with the VO's VOMS, which digitally signs it for future validations.

In this model, users always push all attributes necessary for authorization to resources; in other words, resources never directly pull attributes from VO or institutional repositories on behalf of the user. When AC-enhanced credentials are pushed to a resource, the resource gateway extracts all user attributes and conveys them to a repository of authorization policies, or Policy Decision Point (PDP), central to the site. In turn, the PDP replies with an authorization decision and a set of privilege constraints, also called Obligations. The gateway acts as a Policy Enforcement Point (PEP) and enforces the PDP decision. Figure 1 shows a diagram of the security model.

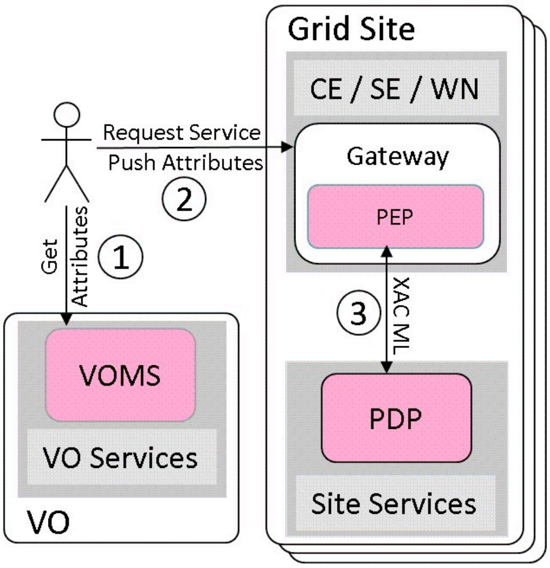

Fig. 1: A diagram of the OSG and EGEE Security Model. User attributes are obtained from VOMS (1). Service requests are issued by pushing user credentials and attributes to a resource gateway at a grid site (2). The resource gateway acts as a Policy Enforcement Point (PEP) and contacts the site-central Policy Decision Points (PDP) using the common XACML authorization interoperability protocol (3).

The authorization interoperability activity focused on standardizing a protocol for PEP to PDP communication. Despite the commonality of the security model, this activity was a fundamental step to allow the deployment of resource gateway implementations on OSG or EGEE, without the need for grid-specific authorization plug-ins. The common protocol allows grid developer groups associated with EGEE or OSG to reuse a common implementation of the security call-out libraries, thus reducing maintenance and eliminating duplication of work.

4. The Authorization Interoperability protocol

In the EGEE and OSG security model, authorization is based on user's X509 identity attributes and VO membership attributes. These attributes are all pushed by the user to the resource gateway. The Authorization Interoperability protocol uses the SAML v2.0 profile of XACML v2.0 [16,17] to encapsulate all these user attributes in a common profile [18]. The profile also provides an abstraction for what resource types and what actions are considered within the authorization model.

4.1. The SAML profile of XACML

The Extensible Access Control Markup Language (XACML) is a standard defined by OASIS. It is a core XML schema for representing authorization and entitlement policies. The Security Assertion Markup Language (SAML), developed also by OASIS, is an XML-based framework for communicating user authentication, entitlement, and attribute information. As of version 2, the two standards can be used together to allow for greater power and flexibility.

In the context of this paper, the SAML profile of XACML is used to convey detailed information about the subject, resource, and action from the service gateway (PEP) to the centralized authorization service (PDP) and to convey back the authorization decision and optional local-account mapping information.

In XACML terminology, the message sent from PEP to PDP is referred to as a "Request Context", while the message going from PDP to PEP is referred to as a "Response".

Note that for our work we have only standardized on the authorization query interface as specified by the SAML-XACML profile, and do not mandate any use of XACML-compliant policy evaluators at the PDPs.

4.2. Attribute Namespace

A SAML profile can accommodate several profile extensions, so to avoid conflicts with other extensions, the Authorization Interoperability profile uses its own prefix; "http://authz-interop.org/xacml" in URL format and the associated "x-urn:authz-interop:xacml" in URN format.

Attributes can use either the URL or the URN formats. While this group prefers the URL style, we acknowledge that both styles present advantages and disadvantages.

The URL style namespace is preferred because it does not require the registration of a namespace with any standardization body. The uniqueness of the namespace is derived by the uniqueness of the domain name. Moreover, additional services for XML schema resolution and location can be established at the registered domain. For example, both OGF and W3C support direct mapping and resolution of registered XML infoset schemas into URLs.

The URN style namespace is instead desirable for reasons of compatibility with standards bodies like OASIS and IETF; however, using a URN requires the formal registration of the namespace with bodies like IANA. To obviate this problem, the Authorization Interoperability profile defines a URN starting with the "x-" prefix, for "experimental namespace", that doesn't require registration with IANA[19].

4.3. XACML Request

In the XACML model, the PEP sends an XACML access authorization request to the PDP on behalf of a user or a service (the "subject"), to execute an "action" on a "resource" controlled by the PEP, with certain request conditions or "environment", like the time of the request. The request, therefore, contains four attribute sections or contexts ("subject", "action", "resource", "environment") that define its scope. We discuss below these four contexts in more detail.

- "**Subject**" - A PEP uses the subject context to declare for what entity the authorization decision is requested. The subject attributes are used to determine an authorization decision, but not all attributes in the subject section need necessarily to play a role in the decision.
- "**Resource**" - The attributes in the Resource context describe the resource targeted for the authorization request. The resource is typically under the control of the PEP, which acts as a gateway to the resource.
- "**Action**" - The attributes in the Action context describe what action the subject wants to perform on the specified resource.
- "**Environment**" - The attributes in the Environment context convey additional parameters in the authorization request of the subject to perform an action on the specified resource. Sometimes, these attributes specify conditions like the time of the request, but profiles, like the Authorization Interoperability profile, use it for more complex use cases, as discussed below.

4.4. AuthZ Interop Request Profile

The Authorization Interoperability group has agreed on a profile for additional XACML request attributes, on each of the four XACML request contexts. These attributes encapsulate the access authorization use cases common to the OSG and EGEE models. The following is a short summary of the profile attributes, organized by context. The reader is encouraged to read the document "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids"[18] for a full description of the attributes and examples of XACML messages and policies.

Subject: Both OSG and EGEE authorization infrastructures define user and service identities via X.509 proxy certificates, extended with VOMS attributes to encapsulate user attributes, like Virtual Organization membership. In our profile, the attribute namespace for the subject context is "http://authz-interop.org/xacml/subject/". Within this namespace, the XACML attributes related to basic X.509 properties are:

- subject-x509-id: the Distinguished Name (DN) of the user or service requesting the access authorization.
- subject-x509-issuer: the DN of the entity that signed the user or service certificate, typically a CA.

The XACML attributes related to VOMS attributes are:

- vo: the name of the Virtual Organization for which the user is requesting the access authorization.
- voms-signing-subject: the DN of the VOMS service certificate that signed the VOMS attribute.
- voms-signing-issuer: the DN of the entity that signed the VOMS service certificate; this is typically a CA.
- voms-fqan: the list of fully qualified attribute names for the subject; the FQANs express the membership of the subject to VO groups and group roles.
- voms-primary-fqan: the first element of the FQAN list; this FQAN carries particular significance in the OSG and EGEE model, since the user specifies this FQAN to define on behalf of what VO group or group role she is doing work for.

One XACML attribute in the profile is used to define a condor canonical name:

- subject-condor-canonical-name-id: in the condor system, privileges are associated to users, identified by canonical names. This attributes carries the user canonical name.

A series of optional attributes, not discussed in this paper, are also defined in the profile.

Resource: in our profile, we define only resources of particular interest to our community. The resource targeted by the request is expressed using the OASIS attribute name "resource-id". The possible values, prefixed with "http://authz-interop.org/xacml/resource-type/", are:

- CE: a computing element is a gateway to a cluster of computing resources; typically, a CE controls access to a computing cluster.
- WN: a worker node is a machine that is part of a computing cluster. This resource is generally controlled by a local batch system and may not be directly accessible by the grid or the grid authorization infrastructure. Both EGEE and OSG, however, adopt pilot-based workload management systems, like GlideinWMS[20], Panda[21] and DIRAC[22], that allow the registration of a worker node to a VO-specific pool of grid resources; this registration is achieved by submitting to the CE a "pilot" job, which is then responsible for the execution of user jobs. In these cases, access to a WN can be centrally controlled by the site authorization system.
- SE: a storage element controls access to files and storage pools.

Other attributes that characterize grid resources are defined within the namespace "http://authz-interop.org/xacml/resource/". These attributes carry information such as the Domain Name of the resource or the DN and issuer of the X.509 host certificates that defines the resource identity.

Action: in our profile, the action is expressed using the OASIS attribute name "action-id". We defined an enumeration of possible actions for "action-id", each used in specific OSG and EGEE use cases. The possible actions, prefixed with "http://authz-interop.org/xacml/action-type/", are:

- queue: this action states that the subject requests authorization to interact with the job queue of the specified computing resource. This action is used in conjunction with the CE resource type, typically when requesting authorization to submit a job to the batch system queue controlled by a CE.
- execute-now: this action states that the subject requests authorization to execute immediately a job at the specified computing resource This action is used in conjunction with the CE (computing element) or WN (worker node) resource types, to execute a job at the computing element resource gateway machine or at a worker node.
- access: this action states that the subject requests authorization to access a specified storage resource. The scope of the request is implementation-dependent: the request can specify access to a single file, a list of files, or a remote/local storage pool. By design, this action generalizes finer-grain types of access, like read access, write access, file system administrative access, etc. Such fine-grain access control is delegated to the authorization layer of storage services.

Since both EGEE and OSG use Globus to control access to computing resources, the profile defines one attribute to convey the detail of the Globus request:

- http://authz-interop.org/xacml/action/rsl-string: the Globus Resource Specification Language string.

Environment: as mentioned in the resource section, both OSG and EGEE support direct management of jobs to Worker Nodes via pilot-based workload management systems. Our profile uses the environment context to convey to the PDP the identity of the pilot job. These attributes use the namespace "http://authz-interop.org/xacml/environment/pilot-job/" and have the same attribute name as the attributes of the subject context.

4.5. XACML Response

In the XACML model, a PDP sends back to a PEP an XACML response, after processing the PEP's XACML request for access authorization.

The principal element of an XACML response is the authorization decision; it can be either "Permit", "Deny", "Indeterminate", or "NotApplicable". In theory, the PEP can query a set of PDPs, and the combined results of those PDPs should evaluate to "Permit" before the PEP will allow access.

If the PDP returns a "Permit", it can also return conditions, known as "Obligations", under which the access can be granted. Obligations typically identify privilege restrictions for the resource access.

4.6. AuthZ Interop Response Profile

The Authorization Interoperability profile defines Obligations to restrict the privileges granted when accessing a computing or storage resource. These privileges are expressed requiring that the PEP grants resource access with the privileges of a specific local POSIX account and/or limits the storage access privileges, e.g. to a specific subset of the file system. The reader is encouraged to read the document "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids"[18] for a full description of the attributes and examples of XACML messages and policies.

The profile uses the namespace "http://authz-interop.org/xacml/obligation/" for its obligations and the namespace "http://authz-interop.org/xacml/attribute/" for the attributes related to these obligations. The obligations defined in the profile are the following:

- uidgid: this requires the PEP to grant access to the resource with the privileges of the specified local Unix ID and Group ID.
- secondary-gids: this requires that the PEP grants the privileges associated to the specified secondary Group IDs.

- username: this requires the PEP to grant access to the resource with the privileges of the specified local Username.
- afs-token: this obligation conveys an AFS token, which the PEP must put in the environment of the resource access process.
- root-and-home-paths: when accessing a storage element, this obligation restricts access to a specified portion of the file system
- storage-access-priority: when accessing a storage resource, this obligation conveys the priority of the request relatively to other requests.
- access-permissions: when accessing a storage resource, this obligation informs the underlying storage system to restrict access to read-only or read-write mode.

5. Implementation

The OSG and EGEE security model have been implemented with different infrastructures by each grid. The following sections describe details of the infrastructure implementations.

5.1. OSG Security Model Implementation

The OSG implementation of the security model is based on the infrastructure provided by the VO Services project [12]. These consist of the PRIMA PEP call-out module and the GUMS PDP [13] for job execution, and the gPlazma call-out module and server [3] for storage.

PRIMA is a plugin based on the Globus Security Infrastructure (GSI) [8]. It extracts the user's X509 Distinguished Name (DN) and the first FQAN in the list of VO membership attributes, if present, and sends them over the network to GUMS. GUMS returns a mapping to a local POSIX account if the user is authorized. It is this local account that implements the restriction of privileges at the resource.

GUMS is implemented as a Web Service around an authorization database. The base GUMS configuration typically consists of a list of VOMS servers and associated mapping rules. On regular intervals, GUMS retrieves the list of user DNs and associated FQANs from all the listed VOMS servers and populates its database accordingly. A request from PRIMA triggers a database search and a mapping is returned if the user's DN and optional FQAN are found in the database. With the new authorization interoperability profile, the mapping information is returned as a "username" obligation.

The communication between PRIMA and GUMS is performed over a GSI connection, with mutual authentication based on X.509 host certificates. Before the authorization interoperability activity, the communication protocol was a modified version of a SAML 1.0 profile [13]. Currently, both PRIMA and GUMS support the authorization interoperability protocol.

A very similar mechanism is used in gPlazma for storage authorization. The gPlazma client uses GSI to extract the user DN and first FQAN, if present, from the X509 proxy certificate, and sends them to the gPlazma server. The gPlazma server forwards this information to GUMS, using the same protocol as PRIMA. After receiving a reply from GUMS, gPlazma augments it with storage-specific attributes and forwards it to the gPlazma client. The protocol between gPlazma client and server is based on Java serialization.

From version 4.2 of the Globus Toolkit, the Globus Web Services GRAM (WS-GRAM) [28] computing gateway natively interfaces its authorization call-out to GUMS using the authorization interoperability protocol and profile. Because of the common protocol, WS-GRAM can also interface to the SCAS PDP.

5.2 EGEE Security Model Implementation

The traditional EGEE implementation of the security model extends the GSI security libraries with the LCAS/LCMAPS framework [10,11]. Authentication and authorization are based mostly on FQANs. LCMAPS uses a modified grid mapfile, which maps the first FQAN in the list to a pool of POSIX accounts. The user DN is not listed in the mapfile, but different DNs are still guaranteed to be mapped to different accounts via an internal tracking mechanism.

Recently, EGEE has recognized the need for a centralized authorization service and has started the implementation of a PDP, called the Site Central Authorization Service (SCAS). A SCAS PEP is also being implemented as an LCMAPS plugin. This plugin is used by common middleware, such as the pre-Web Services Globus Gatekeeper, GridFTP, and the gLExec identity switching tool [4,5]. The SCAS PEP and PDP communicate via the authorization interoperability protocol. Grid to local user mapping information is returned via the "uidgid" and "secondary-gids" obligations. Because of the common authorization protocol, gLExec can be already deployed both in EGEE and OSG with minimal configuration changes.

5.3. XACML libraries

The authorization interoperability activity has developed a set of libraries that implement the Authorization Interoperability protocol. These libraries are used in the implementations of PEPs and PDPs in both OSG and EGEE.

The authorization messages are expressed in XACML format and sent on the wire as SOAP messages over a TLS transportation layer. This protocol can easily be implemented as a web service interface. The XACML message is formed using the OpenSAML v2.0 libraries [23], for the java implementation, and using the Globus Toolkit implementation of the SAML v2 / XACML v2 specifications, for the C implementation.

5.4. Infrastructure Tests and Deployments

After the implementation of the XACML libraries and their integration with the principal resource gateways in OSG and EGEE, the infrastructure has undergone a series of interoperability tests. The targeted resource gateways were the pre-Web Services Globus Gatekeeper, the Web Service Globus Gatekeeper v4.2, GridFTP, the SRM/dCache Storage Service, and the gLExec identity switching tool. Each of these gateways has been tested for authorization against both GUMS and SCAS, with minimal changes to their configuration.

In addition to internal tests, the infrastructure is currently undergoing certification tests in both grids for production deployment. Production deployment is scheduled at dozens of resources for early 2009.

6. Future work

The authorization interoperability collaboration envisions work in three main areas:

1) Extending the support of the protocol to additional resource gateways and policy decision points. These include the Site Authorization Service (SAZ) PDP, Globus Reliable File Transfer and Delegation services (PEPs), and the Berkeley Storage Manager (BeStMan) Storage Service (PEP).
2) Extending the protocol to include additional use cases. These may include additional obligations, especially for the storage use case.
3) Working in the context of the OGSA-Authorization OGF Working Group, making sure that all of our current use cases are included in the more general interoperability activity. The OGF standard should eventually replace this authorization interoperability protocol.

7. Summary

The goal of the Authorization Interoperability activity is to ensure interoperability between the middleware and authorization infrastructures used in the OSG and EGEE projects. Both grids have a common security model, whereby users push to resources identity attributes, based on X509 certificates and VOMS identity attributes. Both grids are also moving toward a distributed authorization infrastructure, with site-central PDPs. In this context, authorization interoperability was achieved by defining a common authorization protocol with an associated profile.

The Authorization Interoperability protocol is based on the SAML v2.0 profile of XACML v2.0. A set of attributes and obligations specific to the needs of OSG, EGEE, Globus, and Condor has also been defined in a profile. The protocol and profile have been implemented as part of the authorization tools of Globus, OSG, and EGEE, while the Condor team is planning to follow suit. Interoperability test suites have helped us to ensure common adherence to the commonly agreed standards across implementations.

The definition of a common protocol is a great step forward for OSG and EGEE, as it enables better interoperability of services as well as providing software reuse opportunities across our projects.

Acknowledgments: Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Dept. Department of Energy. This work was partially funded by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

7. References

[1] R. Pordes, et. al., “The open science grid”, Journal of Physics: Conference Series 78 , Institute of Physics Publishing, 2007 (15pp)

[2] “EGEE Home”, http://www.eu-egee.org/, Accessed October 2008.

[3] A. S. Rana et. al., “Introducing Advanced Fine-grained Security in dCache-SRM for PetaByte-scale Storage Systems on Global Data Grids: gPLAZMA `grid-aware PLuggable AuthoriZation MAnagement System'”, Nuclear Science Symposium Conference Record, 2006. IEEE, pp 632-636, ISBN: 1-4244-0561-0.

[4] I. Sfiligoi et. al., “Addressing the pilot security problem with gLExec”, Journal of Physics: Conference Series 119, Institute of Physics Publishing, 2008 (6pp)

[5] D. Groep et al, “gLExec: gluing grid computing to the Unix world”, Journal of Physics: Conference Series 119, Institute of Physics Publishing, 2008 (11pp)

[6] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection – The Directory: Authentication Framework, June 1997.

[7] S. Tuecke et. al., “Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile”, RFC 3820, http://www.ietf.org/rfc/rfc3820.txt

[8] Overview of the Grid Security Infrastructure, http://www.globus.org/security/overview.html, Accessed October 2008.

[9] R Alfieri et. al., “From gridmap-file to VOMS: managing authorization in a Grid environment”, Future Generation Computer Systems 21 (4) pp549–558 (2005)

[10] R. Alfieri et. al., “Managing Dynamic User Communities in a Grid of Autonomous Resources”, Proceedings of the Computing in High Energy and Nuclear Physics conference, 24-28 March 2003, La Jolla, California, USA (TUBT005, ePrint cs.DC/0306004)

[11] T. Röblitz et al., “Autonomic Management of Large Clusters and Their Integration into the Grid”, Journal of Grid Computing 2 247260 (2004)

[12] VO Services Project Home Page, http://www.fnal.gov/docs/products/voprivilege/, Accessed October 2008.

[13] M. Lorch et. al., “Authorization and account management in the Open Science Grid”, Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing, IEEE, 2005 (8pp), ISBN: 0-7803-9492-5

[14] D. Thain, T. Tannenbaum, and M. Livny, "Distributed Computing in Practice: The Condor Experience", Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pages 323-356, February-April, 2005.

[15] J. Kohl and C. Neuman, “The Kerberos Network Authentication Service (V5)”, RFC 1510, http://www.ietf.org/rfc/rfc1510.txt

[16] “SAML Specifications”, http://saml.xml.org/saml-specifications, Accessed October 2008.

[17] “OASIS XACML TC”, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, Accessed October 2008.

[18] M. Altunay et. al., “An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids”, FNAL Doc DB 2685-v1, Fermilab, 2008 (40pp), http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2685

[19] L. Daigle et. al., “URN Namespace Definition Mechanisms”, RFC 2611, http://www.ietf.org/rfc/rfc2611.txt

[20] I. Sfiligoi, “Making science in the Grid world: using glideins to maximize scientific output”, Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE 2, Honolulu, HI, USA, 2007, pp. 1107-1109, ISBN 978-1-4244-0923-5

[21] “The PanDA Production and Distributed Analysis System”, https://twiki.cern.ch/twiki/bin/view/Atlas/PanDA, Accessed October 2008.

[22] A. Tsaregorodtsev, V. Garonne, and I. Stokes-Rees, “DIRAC: A Scalable Lightweight Architecture for High Throughput Computing”, Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04), 2004, pp. 19-25

[23] Internet2 / OpenSAML: http://opensaml.org, Accessed October 2008.

[24] The OGF OGSA-Authorization Working Group: http://forge.gridforum.org/sf/projects/ogsa-authz , Accessed October 2008

[25] T. Dierks, E. Rescorla, “The Transport Layer Security (TLS) Protocol - Version 1.2”, RFC 5246, http://www.ietf.org/rfc/rfc5246.txt

[26] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, T. Freeman, "A Multi-policy Authorization Framework for Grid Security," pp. 269-272, Fifth IEEE International Symposium on Network Computing and Applications (NCA'06), 2006.

[27] The Site Central Authorization Service information page. http://www.nikhef.nl/grid/lcaslcmaps/scas/ Accessed October 2008

[28] M. Feller, I. Foster, and S. Martin, “GT4 GRAM: a Functionality and Performance Study”, Proceedings of TeraGrid 2007 Conference, Madison, WI.

The following government license should be removed before publication: