A Multipolicy Authorization Framework for Grid Security

Bo Lang,[1,2] Ian Foster,[1,3] Frank Siebenlist,[1,3] Rachana Ananthakrishnan,[1]Tim Freeman[1,3]
[1] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL
[2] Beihang University, Beijing, China
[3] University of Chicago, Chicago
{lang, foster, franks ,ranantha, tfreeman}@mcs.anl.gov

Abstract

A Grid system is a Virtual Organization that is composed of several autonomous domains. Authorization in such a system needs to be flexible and scalable to support multiple security policies. Basing on the Web Services security specifications such as XACML, SAML, and the special security needs of the Grid computing, we have constructed an authorization framework in the Globus Toolkit 4 that can support multiple policies. This paper describes the concepts of our design and introduces the structure and the components of the authorization framework. To show the flexibility and scalability of the framework, we introduce a new blacklist/whitelist-based authorization mechanism that can be seamlessly integrated into the framework.

1. Introduction

Grid is a new kind of distributed computing technology. A Grid system is a virtual organization comprising several independent autonomous domains [1]. Authorization is an important part of the Grid security system. In a grid computing environment, every autonomous domain may have its own policy and may change its policy dynamically. Hence, the authorization mechanism of the Grid system needs to support multiple security policies and needs to have the flexibility to support dynamic changes in security policies, which suggest new challenges to the Grid computing platforms.

With the merging of Grid and Web Services, many new standards and concepts in Web Services are introduced into Grid computing area. Basing on the authorization related specifications in Web Services and the special authorization requirements of Grid, we established a flexible multipolicy authorization framework in Globus Toolkit release 4.

The rest of the paper is organized as follows: section 2 discusses some related work; section 3 introduces the XACML specification, which is the basis of our authorization framework; section 4 describes the design concepts, the structure, and the components of the authorization framework; section 5 discusses the design and implementation of the blacklist/whitelist-based authorization mechanism; section 6 summarizes our work.

2. Related Work

Authorization has been widely studied in the Grid community. In Globus Toolkit, the security functionality is called the Grid Security Infrastructure (GSI) [2,3], and authorization is developing together with GSI. From version 1 in 1998 to the 2 release in 2002 and now the 4 release, GSI has been developing rapidly. In GT1, GSI mainly provided message protection and authentication. In GT2, GSI introduced X.509 proxy certificates to support dynamic creation of computing entities and provided Community Authorization Service (CAS) to implement access control in dynamic created overlaid trust domains. In GT3, the Grid technology worked with the emerging Web services technology. Security functionalities of GSI3 are defined as OGSA(Open Grid Services Architecture) services [4]. In GT4, additional Web Services security specifications are implemented.

Web Services has provided several security standards that have great influence to the Grid computing. XACML (eXtensible Access Control Markup Language) and SAML (Security Assertion Markup Language) are the two important authorization related standards [5].

There are also several authorization systems that support Grid Computing, such as Akenti[6], PERMIS [7], Shibboleth[8], VOMS [9]. Akenti, PERMIS and Shibboleth use user attributes to make authorization decisions; VOMS provides user attributes which can be used for authorization. These authorization systems support their own policies, and can be integrated into GT4 authorization framework as authorization services.

3. The XACML Authorization Model

GT4 implements the WSRF specification. GT4 authorization framework was constructed based on the OASIS XACML and SAML standards [10]. The architecture of the framework uses the XACML authorization model that is shown in Figure 1.

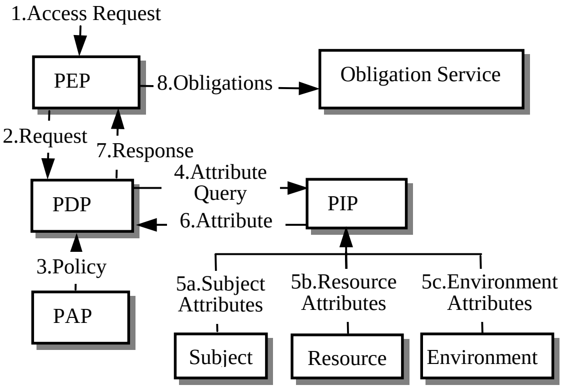

Figure 1. XACML authorization model

The XACML authorization model mainly contains PEP (Policy Enforcement Point), PDP (Policy Decision Point), PIP (Policy Information Point), and PAP (Policy Administration Point). The PEP intercepts the access requests from users and sends the requests to the PDP. The PDP makes access decisions according to the security policy or policy set written by PAP and, using attributes of the subjects, the resource, and the environment obtained by querying the PIP. The access decision given by the PDP is sent to the PEP. The PEP fulfills the obligations and either permits or denies the access request according to the decision of PDP.

XACML also defines a policy language. Policies are organized hierarchically into PolicySets, Policies and Rules, combined using combining algorithms. A rule is composed of a target, an effect and a condition. A Policy consists of a target, one or more rules, and an optional set of obligations.

4. The GT4 Authorization Framework

The convergence of Grid and Web services introduces both new opportunities and new challenges for Grid security. On the one hand, these specifications have provided standard and interoperable methods for Grid security. On the other hand, in order to establish an authorization mechanism suitable for Grid computing, these specifications may also need to be extended or changed to some extent, since Grid has its own special application requirements.

In a Grid system, each domain has its own security policy, such as the grid-mapfile, ACL (Access Control List), CAS, SAML authorization decision assertions, and XACML policy statements. Hence, the GT4 authorization framework needs to support multiple security policies and also needs to be flexible, so that it can be changed easily for different application environments. These special authorization requirements are not addressed in the XACML specification.

Based on the XACML specification and the Grid access control requirements, we designed and implemented the GT4 authorization framework.

4.1. The Framework Architecture

The GT4 authorization framework implements SAML and uses the XACML model, as shown in Figure 2. It is composed of a PEP, PDPs, and PIPs.

For each existing authorization policy, the framework constructs a PDP for evaluating that kind of policy. The Master PDP is responsible for coordinating the PDPs to render a final decision. The Master PDP and the PEP are collectively called the authorization engine. The framework provides different kind of PIPs. A subset of PIP, referred to as Bootstrap PIPs, collect information only about the request, such as the peer subject, the requested action, and the resource. An example of one such PIP, is the X509BootstrapPIP, which extracts the subject DN of the peer from the X509 certificate.

When a request of the Grid resource comes, the PEP intercepts it and sends a decision request to the master PDP. The master PDP collects information needed by calling the Bootstrap PIPs and other PIPs and then invokes the corresponding PDPs with the request and the information collected. The PIPs and the PDPs used are all specified in the security configuration file. When the master PDP receives the decisions returned by each PDP, it combines the decisions, using a policy combination algorithm, such as deny override or permit override, to render a final decision and returns the decision to the PEP. The PEP then executes the decision, either denying or permitting the request.

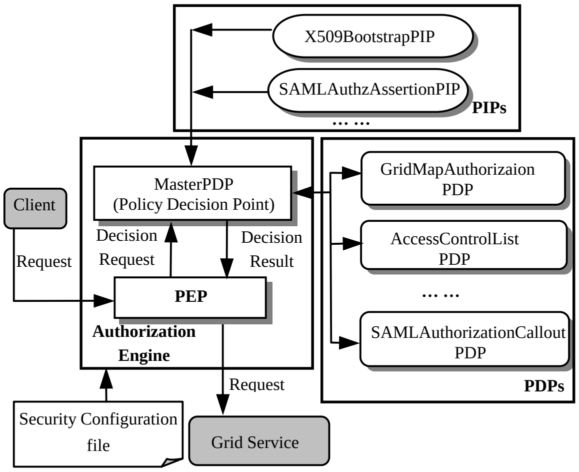

Figure 2. GT4 authorization framework

4.2. The PDP of the Authorization Framework

The PDP is the core of the authorization framework. In order to make the framework support different kind of policies and be scalable, we built a multipolicy framework as shown in Figure 3.

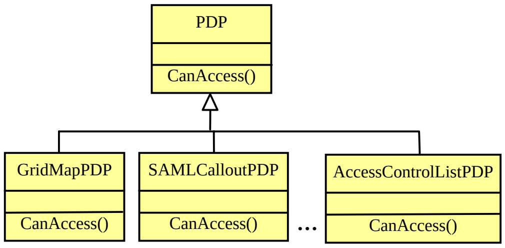

Figure 3. Authorization policy framework

Because every policy essentially needs its own custom decision evaluator that understands the intrinsic semantics of the policy expressions, it is necessary to encapsulate the policy into an independent PDP. At the same time, we abstract the common characteristic of the policies and define an abstract PDP. The PDP abstraction (the PDP class in Figure 3) defines a common interface that can be used to interact with the PEP or with other PDPs. Each specific policy is a subclass of the PDP abstraction, which implements the common interface inherited from PDP with its own policy and evaluation mechanism.

The policy framework is object-oriented. New policies can be added just by inheriting the PDP class, and the existing policies can be removed and modified at any time. Also, since PDP instances are queried through the same interface and the mechanism-specific details of the PDPs are all hidden behind the public interface, a change to the policy framework has no effect on the Master PDP: it can cooperate with any specific PDPs designated by the security configuration files. This multipolicy framework thus provides users with a flexible and scalable authorization mechanism.

In Grid systems, there are several frequently used simple authorization policies or mechanisms, we provided PDPs that implement these existing policies, such as the AccessControlList PDP and the GridMapAuthorizaion PDP. There are also some authorization systems developed by others that can be used in a Grid system, such as Shibboleth, VOMS and PERMIS. Therefore, we established a SAMLAuthorizationCallout PDP for integrating those authorization systems through the SAML assertions.

5. Blacklist/Whitelist Based Authorization

Blacklist and whitelist mechanisms are simple and well known in the security area. The most obvious advantages of this technology are simplicity and efficiency. They can also be introduced into the Grid services access control area for establishing a simple and effective authorization mechanism. If the authorization mechanism detects the requestor on the blacklist or whitelist, it will make an access decision immediately. Based on the blacklist and whitelist concept, we designed and implemented a prototype BlackListPDP and WhiteListPDP under the GT4 authorization framework. The Blacklist/whitelist-based authorization structure is shown in Figure 4.

The BlackListPDP and the WhiteListPDP are inherited from the PDP abstraction introduced in Section 4.2.

The implementation of these two PDPs has two layers: the functional layer and the implementation layer. The blacklist/whitelist access interface, which now contains a member testing method, is defined at the functional layer. The implementation layer contains two levels: the first level is JNDI, which can integrate various naming and directory services and provide a common interface; the second level is composed by different naming and directory services. In our prototype we use an LDAP server to store and manage the blacklist and the whitelist. The URL of the LDAP server is passed to the BlackListPDP and WhiteListPDP through a configuration file.

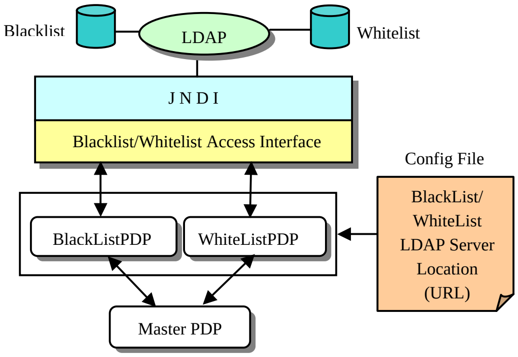
Blacklist
LDAP
Whitelist
JNDI
Blacklist/Whitelist Access Interface
Config File
BlackList/ WhiteList LDAP Server Location (URL)
BlackListPDP
WhiteListPDP
Master PDP

Figure 4. Blacklist/Whitelist-based authorization structure

The blacklist and whitelist are composed of attributes of requestors, such as DN (Distinguished Name, which can be abstracted from the requestor's X.509 certificate), name, and email address. We chose the DN as the identity attribute. Other attributes such as username and group membership can also be used as the identity attributes. This can be achieved by establishing a blacklist/whitelist PIP, which obtains these identity attributes by querying an outside attribute authority using the requestor's DN, and then provides the identity attributes to the BlackListPDP or WhiteListPDP. This will provides more flexibility for users in different application environments.

The blacklist/whitelist-based authorization can also be used together with other authorization mechanisms to make an efficient and rigorous authorization system. The Master PDP will first call the BlackListPDP or the WhiteListPDP; if the requestor is not found here, other PDPs will be called to do further decision making.

6. Conclusion

We have built a flexible multipolicy authorization framework for GT4. The framework is based on the XACML and SAML specifications. The blacklist/whitelist authorization system established under the GT4 authorization framework can provide a simple and efficient method for Grid service access control. Also, this work illustrates that the GT4 authorization framework is open, scalable, and flexible. The framework is still under development. We expect to provide a more stable version in GT4.2 which will be published later this year.

Acknowledgments Work on GT4 GSI has been funded in part by NSF, by IBM, and by the U.S. Department of Energy under Contract W-31-109-Eng-38.

Reference

[1] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications,* 15(3), 2001.
[2] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A Security Architecture for Computational Grids**.** *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.
[3] V. Welch, F. Siebenlist, I. Foster, J. Brresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuedke, Security for Grid Services. *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, June 2003.
[4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Open Grid Service Infrastructure WG*, Global Grid Forum, June 22, 2002.
[5] M. Naedele, Standards for XML and Web Services Security, *Computer*, vol.36, No.4, PP96-98, April, 2003.
[6] M.Thompson, A. Essiari, S. Mudumbai , Certificate-based Authorization Policy in a PKI Environment, *ACM Transactions on Infomation and System Security (TISSEC),* Volume 6, Issue 4, pp: 566-588, November 2003.
[7] D. W. Chadwick, and A. Otenko, The PERMIS X.509 Role Based Privilege Management Infrastructure. *7th ACM Symposium on Access Control Models and Technologies*, 2002.
[8] V. Welch, T. Barton, K. Keahey, and F. Siebenlist. Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration. In *4th Annual PKI R&D Workshop*, April 2005.
[9] R. Alfieri, R. Cecchini, V. Ciaschini, L. Dell'agnello, A. Frohner, A. Gianoli, K.Lorentey , F. Spataro, VOMS, An Authorization System for Virtual Organizations, In *1st European Across Grids Conference, Santiago de Compostela*, February 13-14, 2003
[10] OASIS, extensible Access Control Markup Language (XACML), V2.0, January 2005