Policy-based Access Control Framework for Grid Computing

Jin Wu, Chokchai Box Leangsuksun, Vishal Rampure
Hong Ong[2]
[1]*Computer Science Department, Louisiana Tech University*
Ruston, LA 71272, USA
[2]*Computer Science and Mathematics Division, Oak Ridge National Laboratory*
Oak Ridge, TN 37831, USA
[1]*{box, vdr003}@latech.edu, jinwu_cn@hotmail.com,* [2]*hongong@ornl.gov*

Abstract

Grid technology enables access and sharing of data and computational resources across administrative domains. Thus, it is important to provide a uniform access and management mechanism couple with fine-grain usage policies for enforcing authorization. In this paper, we describe our work on enabling fine-grain access control for resource usage and management. We describe the prototype as well as the policy mark-up language that we designed to express fine-grain security policies. We then present our experimental results and discuss our plans for future work.

Key words: Grid Computing, Access Control, Attribute-based, Security, Middleware.

1 Introduction

A Grid environment provides a powerful virtual system, composed of a large collection of networked heterogeneous computers. This in turn enables institutions of all sizes to share data and computational resources across administrative domains. As such, resource security and integrity are prime concerns. Grid-based tools such as Globus Toolkit [1], Avaki, Data Synapse, and Entropia [4] have addressed the issues of authentication and secure communication in great depth.

Recently, the need to express and enforce fine-grain policies on the usage of resources has increased. For instance, the site administrator may want to specify exactly what fractions of resources may be used by a given entity. The simple access control mechanism in existing tools is not sufficient to meet this requirement. In this paper, we address this particular issue. We present our design of a customizable framework to provide finer-grain resource access control. We implement our design as an extension to Globus Toolkit (GT2) resource management mechanism. We then describe the policy markup language used to express fine-grain service and resource usage policies. Our preliminary experimental results show the prototype implementation has reasonable performance and costs.

This paper is organized as follows. Section 2 describes related work and highlights challenges in existing policy-based access control implementations. Section 3 describes our policy-based access control framework. Implementation issues and experiment results. The paper concludes and indicates future work in Section 6, respectively

2 Related Work and Challenges

"Classified Advertisements" (Class Ad) is the most notable paradigm for achieving authorization and access control in gird computing. Condor [1] and Generic Authorization and Access Control API (GAA-API) [3] are two examples that implemented Class Ad paradigm. Furthermore, GAA-API provides a generic framework where applications aid to determine access control decisions, request authorization information about a particular resource, and evaluate site policy against credentials.

The Virtual Organization Membership Service (VOMS) [13] was developed by EDG and DataTAG for granting access authorization to distributed resources at Virtual Organization (VO) level. Gridshib [10] is a project funded by the NSF Middleware Initiative (NMI), whose goal is to

0-7695-2585-7/06 $20.00 © 2006 IEEE

combine Grid security in the Globus Toolkit [6] and Shibboleth [11] to enable an identity federation between the Grid and higher-education communities. Gridshib is still under development.

Grid Resource Allocation Management (GRAM) is a component of the Globus Toolkit [6]. It provides resource allocation, execution and status management. Prior to a job submission, the remote user must submit a digital certificate, specified in the X.509 format, to the service host. The GRAM gatekeeper, resides on the service host, authenticates the certificate with a private key and maps the remote user to a local user. Once the remote user is authenticated, the GRAM gatekeeper creates a job manager to serve the job execution request. When the job execution terminates, the job manager returns the job status and results to the remote user. It is worth noting there is no further resource access restriction in this paradigm once a client is authenticated to a site. This is known as "all or nothing" paradigm since it is impossible for the site administrator to honor and enforce the control with fine-grained usage policies.

Currently, there is no existing well-defined model to set and examine authentication policies of remote resources for grid job submissions. It is highly desirable to provide system administrators or resources owner the ability to precisely control resource access based on user information and profile.

3 Fine-grain Resource Access Control Architecture

Our approach to support fine-grain access control is to enhance the Globus GRAM architecture. Figure 1 illustrates our enhanced infrastructure and control flow. It is worth nothing that this enhancement does not require any changes to the original X.509 authentication mechanism.

The policy engine, located after GRAM gatekeeper, (see Figure 1), is based on LDAP repository. The policy engine is automatically started after the GRAM gatekeeper has authenticated the clients' certificate. At this time, the policy engine parses the site policy file until a decision is made, i.e., accept or reject a request. The site policy file is written in Policy Markup Language (PML), similar to XACML. The policy engine will continue to resolve users' policies using the appropriate restrictions. Another feature supported by the policy engine is a resource management by interfacing to the local resource managers (e.g. Torque/PBS).

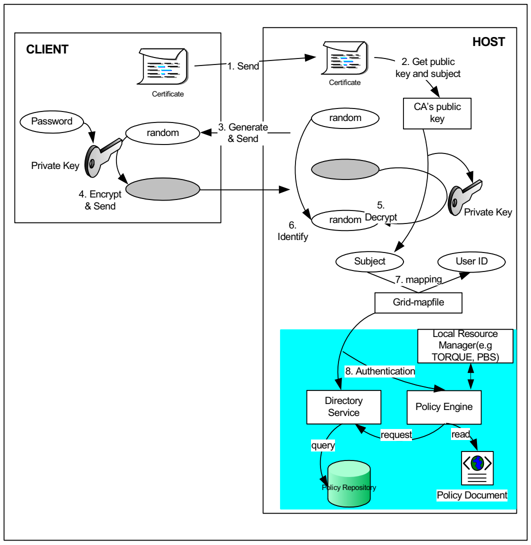

Figure 1: Fine-grain Resource Access Control Architecture

The policy engine automatically resolves attribute-values from an LDAP server when the GRAM gatekeeper receives a job submission requests. For example, if we want to examine the User Type (e.g. GUType) attribute, the policy engine will query the LDAP repository to get its value and compare it to the result specified in policy file to decide depending on the evaluations of the expressions. Original attribute-value pairs in the policy can be modified or deleted and new ones can also be added. Figure 2 depicts our framework in GRAM context.

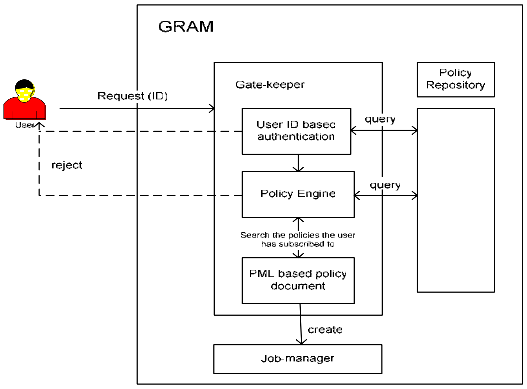
GRAM
User
Request (ID)
reject
Gate-keeper
User ID basec authentication
Policy Engine
Search the policies the user has subscribed to
PML based policy document
query
query
Policy Repository
create
Job-manager

Figure 2: Policy Engine Module in Globus/GRAM Context.

4 Policy Markup Language

We extended the Policy Markup Language (PML), previously created by Gujja and Leangsuksun [5], to express fine-grained access control requirements.

The sample policy file is as shown in Figure 3:

```
<POLICY POLICYNAME=”ACCESS”>
   <CHECK IP=”138.47.*”
TIME=”09:00-17:00”
DAY=”01-20”
MONTH=”01-10”
YEAR=”05-08”/>
   <IF RESULT=”true”
ACTION=”continue”/>
   <ELSE ACTION=”reject”/>
        <RETRIEVE VAR=”DeptName”/>
   <IF RESULT=”CS”
        ACTION=”allow”/>
    <ELSE ACTION=”continue”/>
   <QUOTA JOB=”5/day”/>
   <IF RESULT=”true”
```

Figure 3 an example of our policy file

There are five major tags in the sample policy file.

<CHECK> function is used to check the address (IP address or network address) of the job submission machines, and the access time.

<RETRIEVE> function specifies to query the data repository for the value of the desired attribute. <IF> function is used to check condition to make decision for the policy.

<ELSE> function normally follows the <IF> function. It is used to make decision for the requests which do not meet the requirement in the <IF> function.

<QUOTA> function is used to assign the quantities for the jobs and updating time period.

Further detail on how to extend more tags to provide additional access control functionality can be found in [14].

5 Experiment Result & Benchmarking

We performed a preliminary experiment to test our policy-based access control framework. The hardware testbed is a Linux 2.4.27 platform. The Globus v2.4 environment was used with OpenLDAP-2.0.27 [8] as directory service, and the Berkeley DB version 3.3.11 package [9] harsh database was chosen for LDBM backend. The remote resources were Pentium 4 machines running at 2.4 GHz with 256MB RAM. We measured a job completion time for client requests with and without our enhancement.

To study scalability issue, we varied the number of entries in the LDAP repository from 20 to 10,000 in our experiment. In particular, we examined a submission of three different jobs that represent load variations of 5, 60 and 300 seconds respectively.

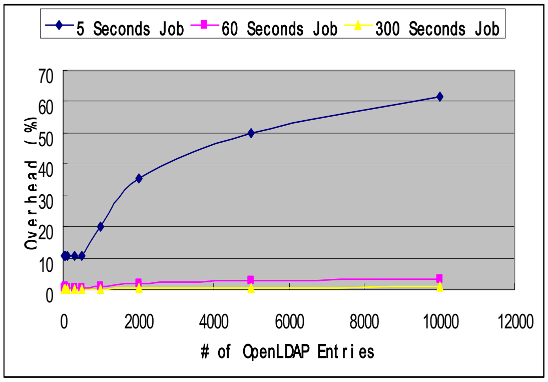

Figure 4 Comparison among various job size based on different numbers of subscribers and policy entries in LDAP Server

Figure 4 shows the performance comparisons. An interesting observation is the impact of the LDAP server is considerable when the job size is small. As the job size becomes larger, the overhead turns out to be very insignificant. For a job running over 60 seconds, the impact is minimal and only depends on the backend database limitation [7].

The experimental results demonstrate that our enhanced architecture enables the fine-gain resource access control for grid job submissions without affecting overall system performance.

6 Conclusion and Future Work

To address the issue of fine-grain resource access control, we have developed a policy-based framework and a policy markup language to precisely control access to resources. Our technique provides a means for the owners and the administrators to enforce fine-grained control and enjoy maximum priority over their resources.

The proposed framework described in this paper provides a fine-grained access control to system resources for grid environment. The policy engine retrieves users' attributes from the LDAP repository for authentication and authorization. We anticipate that the user information in the repository may be compromised. Thus, the privacy of users must be maintained, especially when a grid spans over multiple institutions or organizations. The basic architecture with the privacy authority is shown in Figure 5.

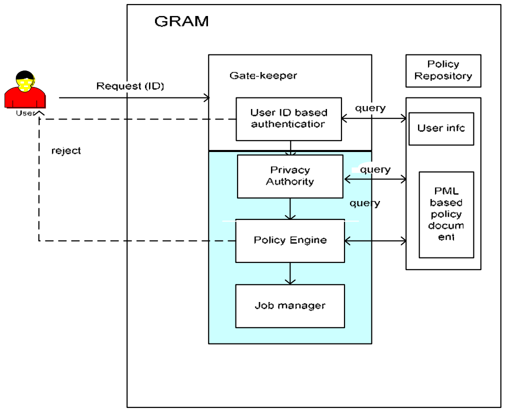

Figure 5 Privacy Authority and Policy Engine Module in Globus/GRAM Context.

We plan to release our implantation to the open source community via a website: `http://xcr.cenit.latech.edu`.

Acknowledgement

This research is supported by Office of science, U.S. Department of Energy, the *fast*OS program, under grant contract no. DE-FG02-04ER4614.

References

1. Globus Toolkits: http://www.globus.org
2. Nicholas Coleman, Rajesh Raman, Miron Livny and Marvin Solomon, "Distributed Policy Management and Comprehension with Classified Advertisements," University of Wisconsin, April 9, 2003.
3. J.L Abad-Perio et al, "*PLAS: Policy Language for Authorizations,*" IBM Research Division, http://citeseer.nj.nec.com/abad-peiro99plas.html
4. Introduction to Grid Computing with Globus,http://www.redbooks.ibm.com/abstracts/sg246895.html?Open
5. Narasimha Reddy Gujja, "*Highly Customizable Linux-Based Access Control Framework For Wireless Points,*" Thesis Presented in Partial Fulfillment of the Requirements for the Degree Master of Computer Science, March 2004, Louisiana Tech University.
6. The Globus Project. http://www.globus.org
7. E. J. Thornton, D. Mundy, D. W. Chadwick. "*A Comparative Performance Analysis of 7 Lightweight Directory Access Protocol Directories,*" IS Institute, University of Salford. http://www.terena.nl/conferences/tnc2003/programme/papers/p1d1.pdf
8. The Open Source LDAP Suite. http://www.OpenLDAP.org
9. The Berkeley Database. http://www.sleepycat.com10
10. GridShib Project Web Site, http://grid.ncsa.uiuc.edu/GridShib
11. Shibboleth Project, Internet2, http://shibboleth.internet2.edu/
12. David W. Chadwick, Alexander Otenko, "The PERMIS X.509 Role Based Privilege Management Infrastructure", *the 7th ACM Symposium On Access Control Models and Technologies*, Monterey, California, USA ¦nbsp;¦nbsp; *June 03 - 04, 2002.*
13. Virtual Organization Membership Service (VOMS) Web site: http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html
14. Jin Wu, "Customizable Fine-Grained Access Control Framework for Grid Computing", Master Thesis, Computer Science Program, Louisiana Tech University, May 2005.
15. Ramkumar Natarajan et al, "*A XML based Policy-Driven Information Service,*"http://citeseer.nj.nec.com/natarajan01xml.htm
16. Tatyana Ryutov and Clifford Neuman, "*The Specification and Enforcement of Advanced Security Policies,*" Information Sciences Institute, University of Southern California. http://www.isi.edu/gost/info/gaaapi/doc/papers/policy2002.pdf
17. Von Welch, Tom Barton, Kate Keahey, and Frank Siebenlist. Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration. In *4th Annual PKI R&D Workshop,* 2005