A Semantic Access Control Model for Grid Services

Junzhou Luo, Xiaopeng Wang, Aibo Song
Department of Computer Science & Engineering, Southeast University, China
{jluo, xiaopeng, absong}@seu.edu.cn

Abstract

Grid computing is appropriate for supporting cooperative work. Many designers and engineers from different companies or institutions can dynamically form a virtual organization for a given design task. In order to protect each company's sensitive data and services, access control is therefore necessary and important. In this paper, we present a new approach to authorize and administrate access requests, in which the requests are obliged to negotiate with a policy enforcement point in order to gain access to the target Grid service. The new access control model will exploit semantic web technology, and use machine reasoning about the messages and policies at a semantic level.

Keywords: Grid, Access Control, Security, Semantic Web.

1. Introduction

The term "Grid" refers to systems and applications that integrate and manage resources and services distributed across multiple control domains [1]. Pioneered in an e-science context, Grid technologies are also gathering interest in industry, as a result of their apparent relevance to commercial distributed-computing applications [2]. As an important application area of Grid, industrial design work which is computation-intensive and data-intensive can be solved in Grid environment. In this scenario, Grid systems are aimed to operate widely outside the industrial institutions and closed design networks. This means the Grid for cooperative work in design should comprise groups of individuals and associated resources and services united by a common purpose which located within a multiple administrative domains such as enterprises, institutions, sites, etc.

A serious security problem afflicting many Grid systems is their inability to work well in the presence of firewalls [3]. For protecting sensitive commercial data and services, most enterprises and organizations now use firewalls to prevent outsiders making connections except to specific, low-risk or tightly managed services. Although this is a reasonably effective access control method, meanwhile it also prevents legitimate distributed applications from working. This problem is particularly evident for Grid applications, many of which use dynamic collections of resources that require peer-to-peer connectivity and bidirectional event notifications. They use high-performance UDP-based protocols, whose point of origin is easy to spoof, which leads most firewall operators to block them by default. Furthermore, the Grid applications also involve users sending their own code to run on a grid resource [4], which badly undermine the use of firewall to partition networks into trusted and distrusted domains.

A current solution to these problems is to open various tunnels through the firewall to allow the Grid traffic to pass through. However, since many Grid systems allow users to run their own codes or even access the shell, this simply bypasses the firewall, and leaves one's internal network open to abuse by a malicious imposter or a misbehaving application. Not surprisingly, few system administrators are willing to open the tunnels needed by many Grid systems. Recently, Grid developers have started using Web Services [5, 6] as the basis of a new Open Grid Services Architecture [7]. These Grid Services exploit "low-risk" protocols such as HTTP [8] that were allowed through by many firewall administrators. The problem is that malicious users can exploit the same mechanisms, so network administrators will soon have to block even these protocols. This "arms race" between network administrators and distributed application developers (including crackers) makes security more complex and expensive and decreases its overall effectiveness. Worse still, it is likely that a Grid infrastructure or application that works today may stop working at some future time if it depends on a particular firewall administration policy.

To address these issues, we present a new access control method what is called Semantic Access Control (SAC). According to the semantic descriptions of the policies, requests, resources and other entities, SAC uses machine reasoning at a semantic level to determine whether let the requests pass. Compared with traditional firewalls, this method is more dynamic, flexible, heterogeneous and interoperable. These characters are vital to distributed systems especially for Grid. This paper is organized as follows: the next section presents an overview of access control and related works in Grid systems. The fundamentals and concepts of SAC are stated in section 3. Section 4 describes the architecture

of the Semantic Access Control model and its working flow. Finally, section 5 underlines some concluding results and future research lines.

2. Analyses of Access Control

2.1 Classical Access Control

Access control is the mechanism that allows owners of resources to define, manage and enforce access conditions applicable to each resource [9].The basic concepts upon which the access control model is based determine the flexibility of the model to adapt to different environments and systems. Several access control models have been developed based on different schemes. It is important to realize that the existing classical access control models were developed for closed environments. Consequently, they are built on the basis of modeling the environments that motivated their development. These models include Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-based Access Control (RBAC).

Discretionary Access Control (DAC) was designed for multi-user databases and systems with a few, previously known, users. Changes were rare and all resources were under control of a single entity. Access controlled based on the identity of the requestor and on access rules stating what requestors are (or are not) allowed to do [10].

Mandatory Access Control (MAC) had its origins in military environments where the number of users can be quite large, but with a static, linearly hierarchic classification of these users. The model is based on the definition of a series of security levels and the assignment of levels to resources and users. MAC policies control access based on mandated regulations determined by a central authority [11].

Role-based Access Control (RBAC) is inspired in the business world. The development of RBAC coincides with the advent of corporate intranets. Corporations are usually hierarchically structured and access permissions depend on the position of the user in the hierarchy. The roles are collections of entities and access rights grouped together based on different tasks they perform in the system environment [12]. The main problem with role based access control is that the mechanisms are built on three predefined concepts: "user", "role" and "group". The definition of roles and the grouping of users can facilitate management, especially in corporate information systems, because roles and groups fit naturally in the organizational structures of the companies. However, when applied to some new and more general access control scenarios, these concepts are somewhat artificial. A more general approach is needed in these new environments.

2.2 Grid Access Control Solutions

In the case of classical closed centralized systems, the same entity is responsible for the assignment of attributes or privileges to clients (Authorization) and the evaluation of the access requests to determine whether they must be granted or not (Access Control). All the information required to analyze and evaluate the privileges is stored and managed locally in the same system where the resources reside. When it came to the Grid, this scheme is not compactable, because each virtual organization (VO) [13] and traditional organizations (TO) will have their own policies for access control. These policies and the access control based on them are high dynamic and heterogeneous or even conflicting. And it is not reasonable to expect that heterogeneous systems for different purposes and under control of different parties will be able to define a common homogeneous set of authorization criteria.

This section deals with several today's grid solutions for access control. We present three common grid computing environments, Condor [14], Legion [15] and Globus [16], and we explain why their solutions have to be extended with further functionality for grid applications with semantic web technology.

Condor is a specialized workload management system for compute-intensive jobs. Its resource management mechanisms are similar to UNIX style access control. Main differences are some additional modes of access besides traditional read and write permissions. Also the classical meanings of the read and write permissions are slightly changed, since all permissions are not relative to files, but to the Condor system itself. Therefore read access means the permission to access information from Condor, write to change settings etc. This mechanism is evolved directly from centralized systems and is only applicable to local closed grid environments.

Legion project uses an object oriented approach to grid computing. Thus resources, such as files, services and devices are considered as objects and access to those are through functions of these objects. The Legion access control approach is that each object is responsible for enforcing own access control policy. Thus each object has a MayI function, invoked before any other functions of the object may be called. This function allows resource owners to define their own access control mechanisms. A default MayI implementation exists that is based on ACL's (access control list) and credential checking (authentication). This access control method is more flexible than that in Condor, and is suit for distributed dynamic environments. But it concerns too much about the resources, the VO's policies and the relationships between each resource are omitted.

Globus grid toolkit (GT) proposes mechanisms for translating users' grid identities into local identities and allows users' certificates be delegated cross many

different sites. This would allow users to sign onto the grid and then use resources without further authentication (single sign-on). The benefit for local resource providers would be that through the translations mechanism, local access control policies can be enforced. Globus toolkit version 3.x (GT3) has taken almost everything in Grid environment as services including the security. These security services involve publishing service security policy so that clients can discover dynamically what credentials and mechanisms are needed to establish trust with the service, specifying standards for the exchange of security tokens to allow for interoperability, etc. Globus uses sophisticated hosting environments to handle access control for applications and allow security to adapt without having to change the application. This means the services hosting environments are responsible for both access control and computing, the heavy burthen will inevitably decrease the efficiency of the whole Grid system. Another serious problem is that the service requesters would contact with the resources directly, this could leave the resources in the high risk of be intruded.

From the introduction of classical and grid access control solutions above, we can see that current access control methods lack systemic partition of the entities involved in the access request scenario and omit the interactions between these entities. So they can't work well in the Grid environments which need integrate and balance each entity's security requirements dynamically. On the other hand, with the rapid development of Web Services and Semantic Web technologies, there have been significant advances in building infrastructure suitable for supporting dynamic interactions between clients and services over the internet. As the access control in today's Grid environments also needs these dynamic interactions on edge, we can fully use the Semantic Web technologies to improve access control in service-oriented, open heterogeneous Grid environments. We deal with these issues in the next paragraph.

3. Semantic Access Control for Grid Services

As a Grid system's interoperation is concerned, its participators mainly include the computing resources (Grid Services) and the access requester for the services. If the access requests conform to all the security policies and resources' attributes, it can be executed as expected. The policies mentioned here include the policies of the resources, the policies of the traditional and virtual organizations, etc. Otherwise, if the attributes of the request and resources are in conflict with one or more policies, the access request will be denied by the access control system. So the use of Semantic Web technology in our Semantic Access Control mainly consists in the following four courses:

(i) **Semantic Description of Entities** Describe the semantics of all the entities involved in the access control scenario, especially the semantic of the request and the resources be requested.

(ii) **Semantic Description of Policies** All the policies should be described or presented in semantic way which would require a universal policy markup language to be used. As in heterogeneous distributed Grid environment, it is unrealistic to use only one policy language, so the necessary translations are demanded.

(iii) **Semantic Reasoning and Conflicts Resolving** Based on the semantic descriptions of all the entities and policies, an efficient semantic reasoning mechanism should be taken to get the last result. This mechanism must also have the capabilities of identifying conflicts between polices and resolving them. In addition, it can reason about the current context of the process and what are the allowed interactions given that context.

(iv) **Certification Authentication** The different semantic attributes of the entities and polices should be certified by an authorization entity which is called certification authentication center (CAC).

3.1 Semantic Description of Entities

The entities involved in the Grid request scenario mainly include the request, resources, environment, subjects, etc. In this paragraph, we focus on semantic descriptions of the two most important entities, the request and the resources (Grid Services). For the request, its semantic description would be attached with itself as a request's access certification (RAC). This certification not only contains the identity of the requester but also states the basilic attributes of the request, such as which services and resources be requested, what the actions will be performed and the valid request period. An example of the components of this certification is shown in Fig.1.

A certificate's signature is needed in order to guarantee the integrity of the metadata of the request and the authenticity of the requester's identity.

The semantic description of the Grid services should be based on the relevant properties of the resources. And the dynamic policy allocation relies on a rich set of metadata about the resources. We use resource specification documents (RSD) to describe these properties, such as the resource's location, state, owner and other instant parameters. Different from the web service description language (WSDL), the attributes described in RSD are more dynamic and detailed. Another difference is that WSDL is used directly by the web service requester remotely and the RSD is used by the SAC system locally. So there's no need for a certificate's signature be combined in RSD. Some

ontology of the resources will be used in RSD so that the SAC system can have a full realization of each resource's status and security requirements in time.

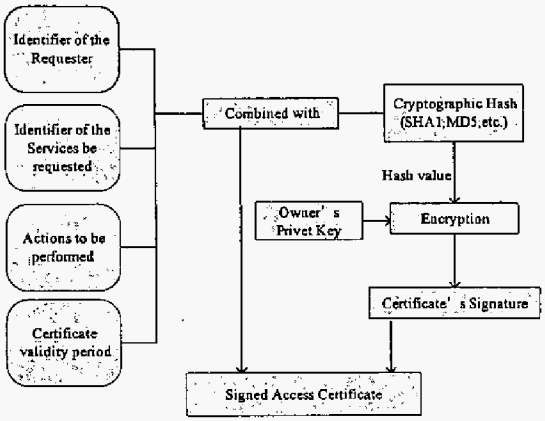

Figure 1.Components of Request's Access Certification

3.2 Semantic Description of Policies

The access control criteria includes different components that participate in the definition of access control policies and can include locally defined components as well as imported elements. All the policies from the TO, VO and resources are needed to be described and analyzed. We use Semantic Policy Language (SPL) which is extended from OASIS's extensible access control markup language (XACML) [17] to define these policies.

As is shown in Fig.2, an SPL Policy is composed of a set of access rules, each one defining a particular combination of a target, an effect and a condition. The target defines the set of resources, subjects, environment to which the rule is intended to apply. The rule combining specifies the procedure by which the results of evaluating the component rules are combined when evaluating the policy. The effect of the rule indicates the rule-writer's intended consequence of a "True" evaluation for the rule. Two values are allowed: "Permit" and "Deny". Condition represents a Boolean expression that refines the applicability of the rule beyond the predicates implied by its target. Therefore, it may be absent. Optionally, operation elements can be used to define which operations of the target resource are controlled by the declared policy, allowing a finer grained access control. In case no operation element is included, the policy is applicable to all of the resource operations. The instantiation element describes the mechanism to instantiate parameters in the policies.

Commonly, each of the SPL policies is attached with a certificate to assure its authenticity and reliability, and some signature technology will be used.

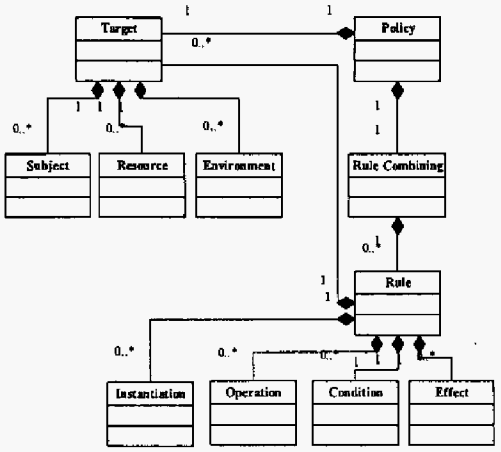

Figure 2.Semantic Policy Language Model

3.3 Semantic Reasoning and Conflicts Resolving

After all the entities and policies in the access control scenario have been semantically described, an efficient semantic reasoning mechanism is required to analyze them in order to deduce an access permit decision.

The usual way for reasoning is policy combination, which is combining several policies into a policy set. For instance, in a personal privacy application, the owner of the information may define certain aspects of disclosure policy, whereas the enterprise that is the custodian of the service may define certain other aspects. In order to render an authorization decision, it must be possible to combine the two separate policies to form the single policy applicable to the request. Then we can simply compare the entities' attributes with this policy set and find whether they are compatible. The rule-combining algorithm defines a procedure for arriving at an authorization decision, given the individual results of evaluation of a set of rules. Similarly, the policy-combining algorithm defines a procedure for arriving at an authorization decision given the individual results of evaluation of a set of policies. Standard combining algorithms are defined for: Deny-overrides, Permit-overrides, First-applicable and Only-one-applicable. In the case of the Deny-overrides algorithm, if a single encountered that evaluates to "Deny", then, regardless of the evaluation result of the other policy elements in the applicable policy, the combined result is "Deny". Other three algorithms are likewise.

Meanwhile, the reasoning logic should also identify conflicts among these polices and try to resolve them. The mechanism for such resolving resolution can vary widely, from the use of pre-established conflict resolution rules to negation over the points in conflict.

A crucial issue is that any resulting solution must be accepted by a re-examination of the policies be each party.

3.4 Certification Authentication Center

The certification authentication center (CAC) is used by SAC to authenticate the certificates of the distributed entities and policies which are specified in XML documents format. In order to finely process these XML documents, CAC would use the newly emerged open W3C XML Key Management Specification (XKMS) [18] to distribute and register public keys.

4. The Semantic Access Control Model

Based on the basis of SAC presented before, we designed a new semantic access control model for Grid services. This model can be implemented as a semantic firewall or something alike in the local network of the Grid services providers. It can protect the resources dynamically according to the semantic attributes of the request, subjects, environment, services and policies.

In this model, we use a policy enforcement point (PEP) to actually execute the access control for the requests, and set a policy decision center (PDC) behind to do semantic reasoning and make access decisions. The policies of TO and VO are collected and managed in the policy administrator center (PAC) and can be submitted to PDC when needed.

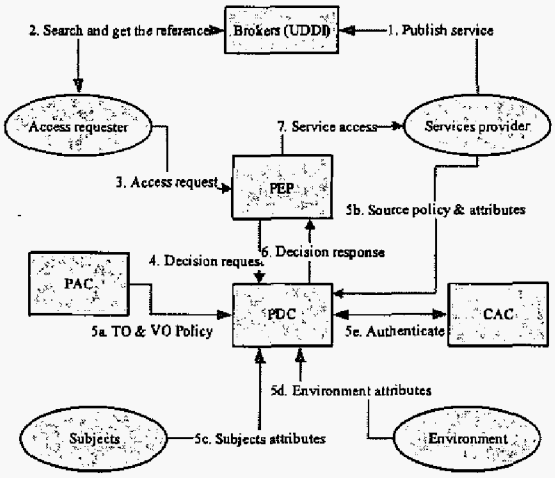

Figure 3.Semantic Access Control Model

Fig. 3 shows the architecture of the semantic access control model and its work flow. We divide the whole process of an access into seven steps. Firstly, the services provider publishes the Grid services to the brokers (UDDI). Then in the second step, the service requester searches the brokers to get the reference (usually an URL) to the service he needed. Thirdly, the access requester sends a request for access to the PEP according the reference, and the access certification is included in this request. In the fourth step, the PEP sends the request for access to the PDC in its native format optionally with the attributes of the subjects, resource and environment. Following this, in the fifth step, the PDC parsed the request, check the certificate cooperated with the CAC and find what entities are involved in it. Then the PDC requests more attributes from the subjects, environment and resource together with policies from the PAC and services. The CAC is contacted again to check these policies' reliability and some translation may be carried out to formalize all policies to SPL language. In the next step, the PDC uses machine reasoning about all the semantic elements gathered to deduce the final decision. During this period some conflicts may occur and the PDC will try to resolve them. Just as the sixth step shows, if the conflicts are irresolvable or the entities attributes are not compatible with the combined policy set then the PDC returns "Deny" to the PEP. Otherwise, it returns "Accept". Then, in the last step, PEP permits the access to the service if the PDC returns "Accept".

5. Conclusions and Future Work

In this paper, we present a new access control method for Grid computing named semantic access control (SAC). It is based on the semantic web technologies in which all the security elements' attributes are semantically described by XML documents in special formats. Based on these formalized policy languages and attributes specification documents, machine reasoning is easily performed to make the access permit decision. Compared with classical access control methods and Grid access control solutions which already exist, this method is more scalable, more applicable to different environments with heterogeneous and complex access criteria and avoids the need of a registration phase. So it can perfectly meet the Grid computing requirements. Further more, the facilities implementing the SAC model can work corporately with traditional firewalls on traditional network edges in a higher network layer. It offers a cheap way for organizations to apply their Grid systems for cooperative work without entirely modifying or weakling their underline network security systems. In conclusion, the SAC model gives a new resolution for easily and flexibly controlling access for Grid services, and it effectively solves the problem of the "arms race" between network administrators and distributed application developers.

There are still a lot of problems to be solved in order to put our SAC model into full implementation in large scope. A universal ontology in Grid security domain is required, more efficient and flexible policy combination algorisms are needed to be designed, etc. The

researches on them are included in our future work plan. In addition, Grid computing and the Semantic Web are now the most rapid developing branches in the Internet research domain. The continuous emerging research achievements in these two branches will definitely give us good illuminations to improve and perfect our SAC model for Grid services.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grants No. 90412014 and Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201.

References

[1] I. Foster, C. Kesselman, *Computational Grids*. I. Foster, C. Kesselman, Morgan Kaufmann,*The Grid: Blueprint for a New Computing Infrastructure*, 1999, 2-48.

[2] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, *Globus Project*, 2002. http://www.globus.org/research/papers/ogsa.pdf.

[3] M. Surridge, C. Upstill, Grid Security: Lessons for Peer-to-Peer Systems, *Proceedings of the Third International Conference on Peer-to-Peer computing*, IEEE, 2003.

[4] V. Roth, Empowering Mobile Software Agents, *Proceedings of the 6th IEEE Mobile Agents Conference*, IEEE CS Press, Los Alamitos, Calif., pp. 238-244, 2002.

[5] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H. Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework", *W3C Working Draft*, June 2002.

[6] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, Web Services Description Language.

[7] OGSA Working Group, "The Open Grid Architecture, Version 1.0", *GGF Working Draft*, http://forge.gridforum.org/projects/ogsa-wg.

[8] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", *RFC 2616*, June 1999.

[9] P. Samarati and S. de Capitani di Vimercati, "Access control: Policies, models, and mechanisms", *In FOSAD 2000, volume 2171 of LNCS*, pages 137 - 196, Springer-Verlag, 2001.

[10] B. W. Lampson, "Protection", *Computer Networks*, 8(1):18 - 24, 1974.

[11] X. Qian, T. Lunt, "A mac policy framework for multilevel relational databases", *IEEE Transactions on Knowledge and Data Engineering*, 8(1):1 - 14, 1996.

[12] D. Ferraiolo, D. Kuhn, "Role based access control", *15th NIST-NSA National Computer Security Conference*, 1992.

[13] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of High Performance Computing Applications*, 15 (3), 200-222, 2001.

[14] Condor Team of the University of Wisconsin, "Condor, high troughput computing", http://www.cs.wisc.edu/condor/.

[15] Legion Research Group of the University of Virginia, "Legion, a worldwide virtual computer", http://legion.virginia.edu/.

[16] Globus Project: Globus toolkit, http://www.globus.org/.

[17] M. Tim, Entrust, "eXtensible Access Control Markup Language, Version 2.0", *OASIS Committee Draft*, September 2004.

[18] F. Warwick, H. Phillip, F. Barbara, "XML Key Management Specification", *W3C Note*, March 2001,http://www.w3.org/TR/xkms.