Enhancing Security Policy Negotiation in the Grid

Lazaros Gymnopoulos[1], Vasilis Tsoumas[2], Ioannis Soupionis[2], and Stefanos Gritzalis[1]

[1] ICSS Laboratory, University of the Aegean, Samos, Greece
[2] ISDB Laboratory, Athens University of Economics and Bussiness, Athens, Greece
e-mail: {lazaros.gymnopoulos, sgritz}@aegean.gr, {bts, jsoup}@aueb.gr

Abstract

The Grid is a major step towards achieving coordinated resource sharing and problem solving within and among virtual organizations. Grid's decentralized nature along with the complexity posed by distributed computation set new challenges for security administrators. In this paper we argue that in order to enhance security management in the Grid, and thus provide answers to the aforementioned challenges, we need to look security policy negotiation from a generic perspective. To do so, we delve deeper into the security policy notion and discover the importance of taming security policy semantics and using uniform policy representations. We present the Security Policy Ontology (SPO) notion along with basic SPO design criteria. Finally, we use the SPO notion to construct a generic framework that enhances security policy negotiation in the Grid; we exemplify using two simple security policies.

Keywords

Grid, Security, Security Policy, Policy Negotiation

1. Introduction

Information systems are evolving from static, geographically confined and isolated "information islands" to dynamically formed, geographically dispersed "information spaces" that are fully interconnected; such "information spaces" are usually referred to as *virtual organizations (VOs)*. The Grid infrastructure (Global Grid Forum, 2004) is a major step towards achieving coordinated resource sharing and problem solving within, and among VOs. In order to achieve these goals, Grid manages intrinsic complexity by defining various abstraction layers (Foster *et al*, 2001).

Security management and configuration takes place throughout those layers, and thus complicates the job of security administrators. In the lower Grid layers security interfaces exist between local systems (local security policies) and the Grid (global security policies). Matching local security policies to Grid security policies poses another important security challenge. While existing security policy conflict resolution or reconciliation frameworks have been applied within specific Grid architecture layers (Wang *et al*, 2004), - with emphasis being given in the lower, more concrete ones - little attention has been given to generic security policy negotiation frameworks. Such a framework could address the security policy management problem throughout Grid abstraction layers, from the more concrete to the more abstract ones.

Before introducing our proposed security policy negotiation enhancement framework, we present an overview of Grid security challenges and requirements in the following section, accentuating the generic perspective of the Grid security policy negotiation problem. In section 3, we make clear that security policy is a multi-interpreted notion and that various ways exist for representing security policies. We base upon this diversity in order to define the fundamental attributes for our framework: manipulation of security policy semantics and uniform representation of security policies. In section 4, we analyze the SPO notion and provide some basic SPO design criteria. In section 5, we present a high level, generic framework for the enhancement of security policy negotiation in the Grid. Finally, in section 6 we conclude and present future work along with open research issues.

2. Grid security challenges and how they form a policy negotiation problem

As a revolutionary technology, the Grid poses new security concerns, not so much in terms of the appearance of novice threats but in terms of the need for increased intensity and flexibility of security mechanisms (Jackson *et al*, 2001). In this perspective one can argue that although Grid incorporates known security challenges and requirements still it introduces some new ones. Gymnopoulos et al present an overview of those challenges and requirements in (Gymnopoulos *et al*, 2003).

In general, security challenges in the Grid can be classified in three categories: *integration* with existing security architectures and models implemented across platforms and hosting environments, *interoperability* of multiple domains and hosting environments at protocol, policy and identity level, and *establishment of trust relationships* among the participants in a Grid system. Meeting the abovementioned security challenges, and thus effectively managing and configuring security, is a much tougher problem in the Grid than it is in classic distributed computing. This is due mainly to three characteristics of Grid computing: *dynamicity*, *autonomy* and *common goal*.

First, the formation of a VO is an entirely dynamic procedure. New resources may become available for sharing at any given time (e.g. if redeemed from another computation) just as new computational needs may occur (e.g. intense need for CPU cycles during a large scale simulation). Second, the lack of central control allows each entity to pursue its own security objectives. Thus, the security problem is upgraded from protecting “the good from the bad” to “reconciling different security perspectives”. Finally, despite central control is absent, coordinated sharing and problem solving is still a Grid objective. Thus, above described advanced security manipulation must be, in the general case, consistent with the need for specific qualities of service (QoS).

The above analysis indicates that generic security policy negotiation mechanisms are vital for managing security in the Grid. Especially the analysis of the last characteristic indicates that those mechanisms should, in the general case, impose the slightest possible load on Grid transactions.

3. Security policy representation

As Wang et al note “the term ‘security policy’ has come to mean different things to different communities” (Wang *et al*, 2004). Indeed, the term “security policy” is interpreted in entirely

different ways that vary from Höne's and Eloff's practical view of security policy as a "vital, direction giving document" (Höne and Eloff, 2002) to Bishop's formalistic definition: "a security policy is a statement of what is, and what is not, allowed" (Bishop, 2002) and from the systemic approach presented by Kokolakis and Kiountouzis in (Kokolakis and Kiountouzis, 2000) to the specialized definition given by McDaniel and Prakash in (McDaniel and Prakash, 2002).

The existence of various interpretations is rooted in two facts. First, security policy is a context dependent notion (e.g. computer security policy, information security policy etc.) but, also, even in the same context specific kinds of security policies have been developed to meet specific needs (e.g. confidentiality security policies in military environments etc.). Both characteristics are evidence for the abounding, in terms of *semantics*, environment that security policies exist in. Therefore in order to manage multiple interacting security policies - and that is the case of the Grid - one has to manage first their semantics.

Along with various interpretations of the security policy notion, several methods of security policy representation exist. Suggestively we mention two polar views that adopt different scientific paradigms. Kokolakis and Kiountouzis adopt the systemic paradigm to develop a "Metapolicy Development System" (Kokolakis and Kiountouzis, 2000), while, on the other side, Gong and Qian, according to Bishop, adopt the systematic paradigm and propose axiomatic rules for the synthesis of security policies such as the "autonomy rule" and the "security rule" (Bishop, 2002). Beyond the abovementioned approaches several more exist and can be roughly divided in the following categories: Verbal Descriptions (Höne and Eloff, 2002), Modelling (Bishop, 2002), Specification (e.g. in (Damianou *et al*, 2001)), and Formalization (Bishop, 2002) (e.g. in (Trcek, 2000)).

The existence of large number of representation methods leads to the conclusion that security policies, even when being semantically compliant, can be presented in ways that differ substantially in terms of formalism, structure, and hierarchy thus raising obstacles in their reconciliation. Therefore, in order to effectively manage security policy negotiation one has to use uniform or at least compatible representation methods.

4. Security policy ontologies

In the previous paragraph we demonstrated the need to manipulate security policy semantics. An efficient means for achieving this purpose is ontology. Ontology is "*an explicit specification of a conceptualization*" (Gruber, 1993). *Domain-specific ontologies* are used to define the terminology for a group of people that share a common view on a specific domain (Decker *et al*, 1999), effectively supporting knowledge sharing and reuse. Thus, security policies can be represented by the means of a Security Policy Ontology (SPO), which elaborates on the domain of security knowledge. SPOs can be used to describe structurally heterogeneous security policies of different levels of abstraction. Thus, by defining shared and common domain theories and vocabularies, SPOs help both people and machines to communicate in a concise manner, a manner which is based not only on the syntax of security policy statements, but on their semantics, as well. Hereby we present the basic SPO design criteria extending definitions from (Gruber, 1993), in order to adapt to the security policy domain:

- **Clarity.** An SPO should effectively communicate the intended meaning of defined terms. Definitions should be objective. While the motivation for defining a concept might arise from social situations or computational requirements, the definition should be independent of social or computational context.
- **Coherence.** An SPO should be coherent; that is, it should attest inferences that are consistent with the security definitions. At least, coherence should apply to the defining axioms.
- **Extendibility.** An SPO should offer a conceptual foundation for a range of anticipated tasks, and the representation should be crafted so that one can extend and specialize the ontology *monotonically*.
- **Minimal encoding bias.** An *encoding bias* results when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized, because knowledge-sharing entities may be implemented in different representation systems and styles of representation.
- **Minimal ontological commitment.** An SPO should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed (with the exception of compliance to legal requirements, such as Data Privacy Acts in place).

5. A framework for enhancing security policy negotiation in the Grid

In section 3, we made clear that security policy is a multi-interpreted notion and that various ways exist for representing security policies. From the first statement we extrapolated that one has to deal with security policy semantics first in order to achieve effective negotiation of security policies in the Grid. The later statement led us to the conclusion that uniform or at least compatible representations of security policies are a prerequisite for the same goal.

In Figure 1 we depict a basic architectural design for a high-level framework that enhances security policy negotiation in the Grid. The proposed framework incorporates both previously drawn conclusions. In order to achieve effective management and homogenization of policy semantics we use a security policy ontology builder. In order to achieve representation compatibility we use an XML parser. The two steps together lead to an enhanced security policy negotiation.

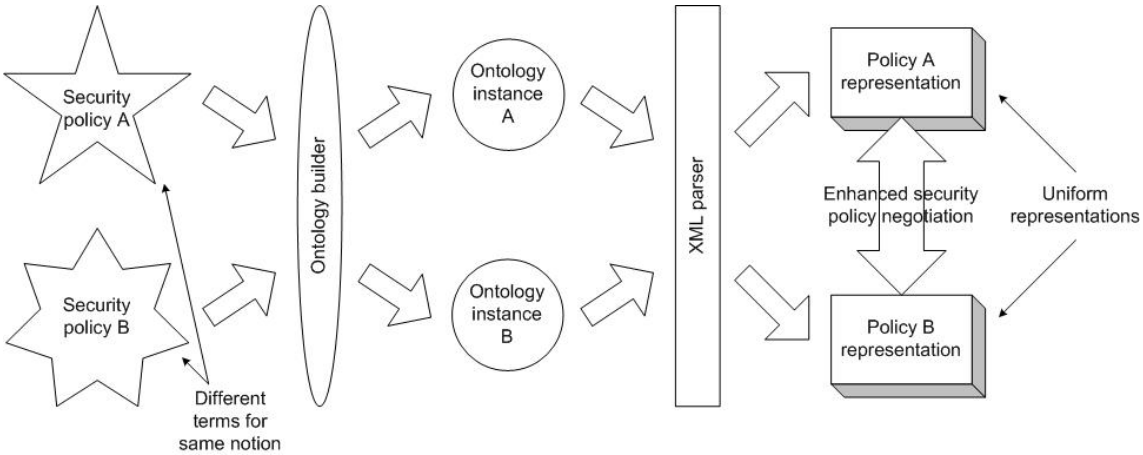

Figure 1 - A framework for enhancing security policy negotiation in the Grid.

In particular, we assume Grid entities (e.g. a resource provider and a resource requestor) that have distinct security policies (security policy A and B respectively in Figure 1) possibly expressed in different languages or even representation models. Both policies are fed to the ontology builder (depicted as an oval in Figure 1). The builder produces a single ontology representation that incorporates notions met in both policies. Each security policy, along with the respective representation model, is then described by a corresponding instance of the aforementioned security policy ontology (ontology instance A and B respectively in Figure 1). The two ontology instances are then used by an XML parser in order to acquire the basic concepts along with their properties and transform them to XML tags and values (policy A representation and policy B representation respectively in Figure 1).

At this point we have to clarify the basic elements of our framework. First, we note that the ontology builder is a semi-automated process that can scan security policy representations and extract security related notions along with their properties and respective values. The builder could handle the existing plethora of different policy representations by using respective interfaces. For example, since most security policy representations follow XML standards, this extraction task may be realized in the above case through XML tools with query support, such as XMLSpy API (XMLSpy API, 2005). Next, the acquired notions can be used for the construction of ontologies which can in turn be merged using one of various existing techniques (e.g. in (Kotis and Vouros, 2004). As an outcome, the ontology builder process constructs several instances of a single security policy ontology that reflect the initial security policies. In this way security policy semantics homogenization is achieved.

Second, the XML-parser (indicated in the same figure as a rectangle) refers to a semi-automated - in the general case - system that is capable of transforming ontology concepts into XML tags and at the same time infusing the proper values to respective attributes. It is noted that the XML parser is fed with data by two different ontology instances created though by the same builder, and thus it can effectively transform inconsistent representations to uniform ones.

5.1 Example usage of the proposed framework

In order to clarify the previously presented framework we provide an example concerning two simple Grid security policies. We assume a simple VO, namely VO A, and a user that wishes to become a member of A and consequently access its resources. VO A and the user have distinct security policies that are represented in arbitrary formats. Here we present both policies in natural language:

User security policy

Authentication
- ***User*** owns a valid pair of ***identification token*** and ***password***
- ***User*** is provided with a *Kerberos* ***ticket***

Authorization
- ***User*** is member of either ***group***: "*Administrator*" or "*Restricted*"

Privacy
- ***Network configuration data*** are *not* ***allowed*** to be transmitted

Virtual organization A security policy

Authentication

Each ***entity*** (*user* or *process*) that wishes to use a ***resource*** must have:
A valid pair of ***ID*** and ***password***
A valid *X.509* ***certificate***
The *X.509* ***certificate*** must be of a *limited duration*
Each ***resource*** must have:
A valid *X.509* ***certificate***
The *X.509* ***certificate*** must be of an *extended duration*
Authorization
Each ***entity*** that uses ***resources*** must have:
A valid pair of ***ID*** and ***password***
Each ***entity*** that uses ***resources*** that belong to ***group*** "*privilege*" must have:
A valid pair of ***ID*** and ***password***
Belong to the ***group*** "*privileged*"
Logging
For each access to the ***resource*** the following data should be logged:
ID, ***password***, and ***IP address*** of the ***entity*** that used the ***resource***

Each policy, irregardless the representation method incorporates some basic security notions according with their attributes. The ontology builder discussed in the previous section has the ability to identify those notions and attributes and successively combine them in a single ontology as shown in Table 1. Some notions, as for example "ID" from the VO security policy and "Identification token" from user security policy, are identified as identical and merged. Others, such as "Resource", exist only in one policy (the VO policy for our example) and still are carried over. Finally, it should be noted, that the structure of each policy is notably different[1] and thus the ontology builder can also shift a notion from one ontology level to the other.

Ontology builder outcome	User security policy	VO security policy
Entity	Null	Entity
Type {User || Process}	Null	Type
Identifier	Identification token	ID
Password	Password	Password
Certificate	Ticket	Certificate
Type {Kerberos || X.509}	Type	Type
Duration {Extended || Limited}	Null	Duration
Group {Administrator || Restricted}	Group	Group
Net configuration	Net configuration data	Null
IP	Null	IP
Allowed {YES || NO}	Allowed	Null
Resource	Null	Resource
Group {Privilege || No Privilege}	Null	Group
Certificate	Null	Certificate
Type {Kerberos || X.509}	Null	Type
Duration {Extended || Limited}	Null	Duration

Table 1: Ontology builder produces coherent security policy ontology

Finally, the ontology builder fills the resultant ontology in order to produce two instances, one for each policy. The XML parser is, then, used to transform the ontology instances to uniform

[1] In Table 1, the VO and user policies are structured in the same way only for representation reasons.

XML representations in order to facilitate automatic security policy negotiation. The final outcome of the framework could be something like the one depicted in Table 2.

User security policy	VO security policy
<?xml version="1.0" encoding="…"?> <Final Policy> <Entity> <Type> </Type> <Identifier> UserId </Identifier> <Password> UserPassword </Password> <Certificate> <Type> Kerberos </Type> <Duration> </Duration> <Group> Restricted </Group> <Net Configuration> <IP> </IP> <Allowed> NO </Allowed> </Net Configuration> </Entity> <Resource> <Group> </Group> <Certificate> <Type> </Type> <Duration> </Duration> </Resource> </Final Policy>	<?xml version="1.0" encoding="…"?> <Final Policy> <Entity> <Type> User </Type> <Identifier> UserId </Identifier> <Password> UserPassword </Password> <Certificate> <Type> X.509 </Type> <Duration> Limited </Duration> <Group> </Group> <Net Configuration> <IP> UserIP </IP> <Allowed> </Allowed> </Net Configuration> </Entity> <Resource> <Group> Privilege </Group> <Certificate> <Type> X.509 </Type> <Duration> Extended </Duration> </Resource> </Final Policy>

Table 2: The framework produces uniform security policy representations.

6. Conclusions and further research

In this paper, we outlined a framework for the enhancement of policy negotiation in the Grid. At present the proposed framework focuses on identification and authentication security policies. We argued that our framework can contribute to the reduction of ambiguity concerning the interpretation of security policies expressed in different ways between negotiating Grid entities. The establishment of a common framework for security information exchange between Grid parties will also provide the foundations for enforcing, evaluating and auditing the security level of Grid security function in a uniform way. Moreover, such a framework will support comparable and reusable axioms between security policies, thus providing a means for semantic queries realization against a Grid policy base.

In this perspective and besides implementing and testing our framework other open issues exist. For example, the way existing security policy reconciliation models can be incorporated in a generic framework, both in the general case and with specific examples, could be examined. Furthermore, an analytical mapping of the proposed framework with Grid architecture layers could be provided. Finally, we plan to investigate how such a framework can be used in order to produce a security policy knowledge management tool for administrators in the Grid.

7. References

Bishop, M. (2002), *Computer Security: Art and Science*, Addison-Wesley, New York.

Damianou, N., Dulay, N., Lupu, E., and Sloman, M. (2001), "The Ponder Policy Specification Language", in Sloman, M., Lobo, J. and Lupu, E.C. (Eds.) *Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, Springer Verlag, Bristol.

Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1999), "Ontobroker: Ontology based access to distributed and semi-structured information", in Meersman R. et al. (Eds.), *DS-8: Semantic Issues in Multimedia Systems*, Kluwer Academic Publishers.

Foster, I., Kesselman, C., and Tuecke S. (2001), "The anatomy of the Grid: Enabling scalable virtual organizations", *International Journal of High Performance Computing Applications*, Vol. 15, No. 3, pp 200.

Global Grid Forum (GGF) Web Site (2004), "The Open Grid Services Architecture, Version 1.0", https://forge.gridforum.org/projects/ogsa-wg/docman/, (Accessed 15 December 2004).

Gruber, T.R. (1993), "Toward principles for the design of ontologies used for knowledge sharing", in *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers.

Gymnopoulos, L., Dritsas, S., Gritzalis, S., and Lambrinoudakis, C. (2003), "Grid Security Review", in Gorodetski, V. (Ed.) *Proceedings of the 2nd International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security*, Springer Verlag, St. Petersburg.

Höne, K. and Eloff, J.H.P. (2002), "Information Security Policy - What do International Information Security Standards say?", in *Proceedings of the 2nd Annual Information Security for South Africa Conference*, Elsevier, Muldersdrift.

Jackson, K.R., Johnston, W.E., and Talwar, S. (2001), "Overview of Security Considerations for Computational and Data Grids", in Bashor, J. (Ed.) *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing*, IEEE Computer Society, San Francisco.

Kokolakis, S.A. and Kiountouzis, E.A. (2000), "Achieving Interoperability in a Multiple-Security-Policies Environment", *Computers & Security*, Vol. 19, No. 3, pp 267.

Kotis K., and Vouros G.A., (2004) "The HCONE approach to Ontology Merging", in Bussler, C., Davies, J., Fensel, D. *et al.* (Eds.) *Proceedings of the 1st European Semantic Web Symposium*, Springer Verlag, Heraklion.

McDaniel, P. and Prakash, A. (2002), "Methods and Limitations of Security Policy Reconciliation", in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, IEEE Computer Society, Berkeley.

Trcek, D. (2000), "Security policy conceptual modeling and formalization for networked information systems", *Computer Communications*, Vol. 23, pp 1716.

Wang, H., Jha, S., Livny, M., and McDaniel, P. (2004), "Security Policy Reconciliation in Distributed Computing Environments", in Chadha, R. (Ed.) *Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks*, IEEE Computer Society, New York.

XMLSpy API, 2005, available at http://www.altova.com/manual2005/XMLSpy/SpyEnterprise/xmlspyapi.htm