A Trust Domain Management Schema for Multiple Grid Environments (Work in Progress)

Vincent C. Hu, Karen Scarfone, Serban I. Gavrila, and David F. Ferraiolo

National Institute of Standards and Technology,

100 Bureau Drive, Gaithersburg, MD 20899-8930, USA

1-301-9754975

{vhu, karen.scarfone, serban.gavrila, dferraiolo}@nist.gov

ABSTRACT

Trust domain management for the global access of a grid is managed under centralized schema for most of the current grid architectures, which are designed based on the concept that there is only one grid for every grid member, therefore requiring central management for authentication and authorization. This design not only has its own limitations, but it is also awkward when a member of a grid may also be a member of other intersecting grids. Schema for such multi-grid environments have not been well thought out. In this paper, we present a schema that enables trust domain management in a dynamic multi-grid environment.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *distributed systems, information networks*.

General Terms

Design, Management, Security, Theory.

Keywords

Access Control, Grid, Policy, Trust Management.

1. GRID TRUST MANAGEMENT

The nodes, members, or computers of a grid are able to act independently without centralized control, but the Trust Domain (TD) (i.e., the coverage of the authentication and authorization for the global access of a grid) is managed under a centralized system for most of the current grid architectures [7, 6, 8]. The current grids have been designed based on the concept that there is only one grid for every grid member, therefore requiring central management in order to authenticate and authorize members. However, in reality, a local member of grid A may want to be a member of grid B – for example, to serve as an information provider for the research community, and at the same time access information from the engineering community, which is unrelated to the research community. Therefore, the multi-grid environment does not have to be in a tree (hierarchical or sub) structure. To accomplish that, a schema is required that is neither based upon the embedded tree architecture nor reliant on the data structure of the access control languages.

Although the layers of current security architecture maintains the independency, authentication, and combination requirements of grid [3], conceals the differences of local security solutions, and provides a unified platform for the upper layers [9], these mechanisms are tied into the data structure of the software or languages they are built from, so they are only available to the TD in hierarchical or sub grid environments. The TD management for such multi-grid environments is adding another layer of protocol to the existing single security architecture.

Centralized TD management would inherit the limitations of centralized systems: 1) a potential single point of failure (i.e., when the root grid fails, its descendent [or sub] grids fail as well), 2) a member joining or leaving a grid requires administration effort from the center TD manager, and 3) a unified protocol and algorithm is required for TD management throughout the whole grid environment. Even though security architecture for grid have been defined [9] that allow expansion for multi-grid environments, there are few schemas that address the limitations. In this paper, we discuss an extension of the TD management paradigm that remedies the issue by creating an algorithm and protocol on top of the current TD management architecture.

2. MULTI-TRUST DOMAIN SCHEMA

To devise a schema for a dynamic multi-grid environment, we need to generate an algorithm and a supporting protocol to handle the authentication and authorization across grids without relying on central TD management or services.

2.1 Trust Domain Algorithm

To support a grid that might be in a non-tree (graph) environment, each grid needs to be able to identify the relations with other grids by referring to the locally maintained directory instead of consulting the central service, and to calculate the access decision according to the provided service requestor (subject) and access rights (capability). The authorization process needs to include a function to find the intersection of TD coverage and determine a remote global access control policy for the combination with the local access control policy of the requested resource [3]. Thus, an access request $<s_x, c_y>$ (i.e., subject s of system x requests to perform capability c of system y) is evaluated for access; if there exists a common global TD for $L(c_y)$ (the local

INFOSCALE'07, June 6–8, 2007, Suzhou, Country.

TD where c_y is covered under) to combine, then there is an access control policy that the access request can be managed under. There may be more than one common global policy that is qualified for the combination, so the least common (common TD that covers the least number of subjects and capabilities) one is best to choose because it has the least number of local TDs involved in the policy integration. Formally, an access request $<s_x, c_y>$ requires $L(c_y) = tdy$ to integrate with global trust domain $LCTD(s_x, c_y) = tdz$ (the least common TD that cover s_x and c_y), and is evaluated by $Grant(P_{tdy}, P_{tdz})$ (the authorization result that is evaluated by the combination of access control policies P_{tdy} and P_{td}) if $CTD(s_x, c_y) \neq \varnothing$ (there is no common trust domain that covers s_x and c_y).

2.2 Protocol Elements

Assuming the protection and format of messages of the underlying security layer are provided, there are various ways to implement the algorithm in Section 2.1 as long as following basic messages are maintained in the TD management protocol:

- The message ($<s_i, c_j>$, $TD(s_i)$) from a local system *li* is sent to the grid network for the access request, where $<s_i, c_j>$ is the access request from subject s_i for resource c_j, and $TD(s_i)$ is stored in *li* and updated when *li* joins or leaves any grid.
- *lj* calculates $LCTD(s_i, c_j)$ and evaluates $Grant(P_{L(c_j)}, P_{LCTD(s_i,c_j)})$ after receiving an access request.
- $L(s_x)$ and $L(c_x)$ of every local *lx* need to be sent to the TDs where they are covered when an update (joining or leaving of a grid member) occurs.
- Every TD should update its coverage list after receiving the updated subjects and capabilities list from the TDs it is managed by and is managing.

The algorithm and protocol described allow the architecture of the multi-grid TD to be built in a graph instead of a tree structure, which permits multiple connections among TDs. This addresses the limitations of being constructed with a single centralized architecture.

3. RELATED WORKS

Grid Security Infrastructure (GSI) is the software developed by Globus [6] staff for use in Globus and other grid applications. It contains a library and a few utilities that are used as a standard mechanism for bridging disparate security mechanisms. GSI's security functions not only understands identity credentials of all grid members but also supports delegation and policy distribution by translating between other mechanisms and GSI as needed and converting from a GSI identity to a local identity for authorization. Multi-grid TD management has not been articulated in SGI; however, the proposed schema can be included as an extension of its current security layer, such that the TD identification is added in the Authentication, Delegation, and Authorization layers (e.g., TD ID extension in X.509 End Entity and Proxy Certification, Attribute assertion in SAML). In contrast to the relatively homogenous approach of GSI, OGSA security envisages translation and mapping of security parameters (e.g., credentials) between different domains [5]. To incorporate our schema in Section 2, the TD information in the protocol should be included in the Identity mapping services (i.e., Trust, Attribute and Bridge/Translation Services Service) such that the combination of the subject DN, issuer DN, and certificate's serial number may be considered to carry not only the subject's or service requestor's identity but also the TD information.

XACML [4] based authorization mechanisms such as Privilege and Role Management Infrastructure Standards (PERMIS) [1], and Shibboleth (with appropriate PDP implementation) [2], which may equip the capability but not yet include the multi-grid mechanism, can also consider incorporating our schema in their authorization functions.

4. CONCLUSION

In this paper, we developed a schema that includes an algorithm and general protocol that handle dynamic multi-grid TD management. The basic idea for the schema is to find the least common TD for the subject and resource of an access request to be combined with the local access control mechanism. Instead of the central management ideas of the existing architectures, the proposed schema relies on the exchange of TD information for each grid, which not only avoids the limitations of a centralized system but also provides the freedom to dynamically participate in grid membership. Although the detailed architecture and message definition are not included in this paper (left for future research), we believe this schema could be used for the next generation of grid TD management design.

5. REFERENCES

[1] Chadwick, D. W., and Otenko, A. The PERMIS X.509 Role Based Privilege Management Infrastructure, *7th ACM Symposium on Access Control Models and Technologies*, 2002.

[2] Erdos, M. and Cantor, S. Shibboleth Architecture v5, *Internet2/MACE*, 2002.

[3] Hu, C. V., Ferraiolo, D. F., and Scarfone, K. Access Control Policy Combinations for the Grid Using the Policy Machine, *Seventh IEEE International Symposium on Cluster Computing and the Grid*, May 2007.

[4] OASIS, XACML TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[5] Robiette, A. Grid Security: Present and Future, *JISC Grid Security Workshop*, Dec. 2002.

[6] The Globus Security Team, Globus Tookit Version 4 Grid Security Infrastructure: A Standards Perspective, www.globus.org/security/overview.html.

[7] The Open Grid Services Architecture, Version 1.0, http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf.

[8] The Open Source Architecture, http://www.apac.redhat.com/software/architecture/forward.php3.

[9] Zhou, Q., Yang, G., Shen, J., and Rong, C. A Scalable Security Architecture for Grid", *Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2005.