Security Requirements Model for Grid Data Management Systems*

Syed Naqvi[1,2], Philippe Massonet[1], and Alvaro Arenas[2]

[1]Centre of Excellence in Information and Communication Technologies (CETIC), Belgium
{syed.naqvi,philippe.massonet}@cetic.be
[2] CCLRC Rutherford Appleton Laboratory, United Kingdom
{s.naqvi,a.e.arenas}@rl.ac.uk

Abstract. In this paper, we present our ongoing work of a policy-driven approach to security requirements of grid data management systems (GDMS). We analyse the security functionalities of existing GDMS to determine their shortcomings that should be addressed in our work. We identify a comprehensive set of security requirements for GDMS followed by the presentation of our proposed Security Requirements Model. Derivation of security policies from security requirements and their consequent refinement is also presented in this paper. Our approach of addressing modelling issues by providing requirements for expressing security related quality of service is the key step to turn storage systems into knowledge representation systems.

Keywords: Grid security, requirements analysis, distributed data management.

1 Introduction

Grids enable access to, and the sharing of, geographically distributed heterogeneous resources such as computation, data and information sources, sensors and instruments, for solving large-scale or complex problems. One of the key Grid applications is the use of grids in emergency response. In this kind of applications, Grids become a critical information infrastructure providing essential information to emergency departments in order to minimise adverse impacts of potential tragedies. For instance, Grids may be useful in preventing floods, which can be achieved by integrating data from various sources - networks of sensors in a river basin, weather prediction centres, historical flood datasets, topography, population and land use data - for processing in sophisticated numerical flood models. The massive data sets that would need to be accessed and processed would require huge network facilities, data storage, and processing power to deliver accurate predictions. This paper focuses on one element of such critical infrastructure: Grid data management systems (GDMS).

We have carried out a formal analysis of security requirements for semantic grid services to explore how these requirements can be expressed as metadata associated to these services. It also explores issues of negotiation of the QoS parameters in order

* This research work is supported by the European Network of Excellence **CoreGRID** (project reference number 004265). The CoreGRID webpage is located at www.coregrid.net.

J. Lopez (Ed.): CRITIS 2006, LNCS 4347, pp. 30 – 41, 2006.

to reach Service Level Agreements (SLA). This work is being used to gridify the *FileStamp* distributed file system which is currently using the peer-to-peer technology for the exchange of data resources across the distributed sites. In this paper, we present a case study of *FileStamp* to explain security requirements model for GDMS.

This paper is organized in the following manner: an overview of the security functionalities of existing GDMS is given in section 2. *FileStamp* distributed file system is presented in section 3. Section 4 illustrates our proposed security requirements model. Our approach vis-à-vis the related work is discussed in section 5. Finally some conclusions are drawn in section 6 along with the outline of our future directions.

2 Overview of Security Functionalities in GDMS

Grid data management systems [1] offer a common view of storage resources distributed over several administrative domains. The storage resources may be not only disks, but also higher-level abstractions such as files, or even file systems or databases.

In this section, an overview of the security functionalities of various existing GDMS is presented:

2.1 ARMADA

Using the Armada framework [2], grid applications access remote data sets by sending data requests through a graph of distributed application objects. The graph is called an *armada* and the objects are called *ships*.

Armada provides authentication and authorization services through a security manager known as the *harbor master.* Before installing an untrusted ship on a harbor, the harbour master authenticates the client wishing to install the ship and authorizes use of the host resources based on the identity of the client and on the security policies set by the host.

The harbor master uses authentication mechanisms, provided by the host machine, to identify clients that wish to install ships on the harbor. The host provides mechanisms that implement security policies set by the host administrator. The options for implementing authentication include using SSH or using Kerberos authentication service.

The most common approaches used to protect system resources from untrusted code are hardware protection (e.g., running the untrusted code in a separate Unix process), software fault isolation (SFI) [3], verification of assembly code [4-5], and use of a type-safe language (e.g., Java or Modula3 [6]). Hardware protection requires untrusted code to run in a separate address space from the harbor. While this clearly protects the harbor from the client code, the overhead of communicating through normal IPC system calls is quite high. Both SFI and verification of assembly code offer promising solutions, but they typically target a limited set of machines, making them non-portable. Type-safe languages provide portability and memory protection for untrusted code: two important features for heterogeneous grid environments.

2.2 GridNFS

GridNFS [7] is a middleware solution that extends distributed file system technology and flexible identity management techniques to meet the needs of grid-based virtual