AUTHORIZATION AND ACCESS CONTROL TO SECURE WEB SERVICES IN A GRID INFRASTRUCTURE

Serena Pastore
National Institute of Astrophysics, Astronomical Observatory of Padova, vicolo Osservatorio 5, 35122, Padova, Italy

Keywords: Grid Security Infrastructure, Authentication and Authorization.

Abstract: Security in a grid infrastructure is implemented by adopting standard protocols that realize authentication, authorization and access control to shared distributed network nodes, resources and services. Despite of middleware used to built a grid, security components provide mutual authentication, delegation and single sign-on features while every virtual organization joining the grid sets own authorization policies. Access control and secure communication are the most important aspect of security that need to be addressed if the shared resources are web services. This paper reports about the strategy required for securing web services as means of an application packaged as a Web ARchive (WAR) file deployed in a grid node that has to be shared for grid users. Software implementation uses packages coming from different past and present grid projects that secure both web container and the application. Security chain is basically implemented by means of java libraries to provide a message handler technique for digital signing and validating SOAP message and an authorization engine compatible with methods adopted in grid.

1 GRID SECURITY

Grid security refers to the need for any Virtual Organization (VO) which composes the grid infrastructure, to share information, resources and even applications across organizational boundaries in a secure and highly efficient manner (Foster, I. et al., 2001). The globally distributed feature of a grid system should allow for all grid entities (users belonging to a VO, nodes and services) authentication to be identified in the infrastructure, secure communication over the network and authorization to make use of shared objects according to both local (site level) and distributed policies (VO level). Enabling remote access to different network systems and sharing resources require many other services like mutual authentication among each parties forming the grid to prove who they are, single sign-on for user and delegation to have the necessary permissions to act on behalf of a grid entity. Access control and secure communication are the most important aspect of security that needs to be addressed for sharing web service application: access control in specific encompasses a number of concepts that includes knowing who users are (identity) and what they can do in the applications (authorization) and keeping record of what they have done (auditing). Despite middleware used to built a grid, components which address security requirements adopt standard technologies. According the Globus Security Infrastructure (GSI) (http://www.globus.org) that is the reference architecture developed by the Globus Alliance, the SSL(Secure Sockets Layer) protocol provides secure communication, authentication is realized by using digital certificate of the Public Key Infrastructure(PKI) (http//www.ietf.org/), while various authorization models are implemented. GSI creates a basic security mechanism that does not require a centralized management authority. Public key cryptography is used for digitally signing any piece of information. Each grid entity has an unique identity composed by a signed certificate following the X.509 format and a private key (credentials) which authenticate it in grid. A Certification Authority (CA) guarantees identities by issuing signed certificates that validate individuals and organizations: in a global grid environment with many parties involved, each CA is structured to form a hierarchy of trusted entities. Delegation and mutual authentication services are provided to grid by a proxy method. This introduces new temporary credentials (a certificate signed by the owner from the original one and a private key with a limited lifetimes) that can be passed to any grid resource. Authorization indeed outlines the core problem in

Pastore S. (2006).
AUTHORIZATION AND ACCESS CONTROL TO SECURE WEB SERVICES IN A GRID INFRASTRUCTURE.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 264-267
DOI: 10.5220/0001252802640267

grid setting that is how to handle and combine the overlay of polices imposed by different organizations. A first method uses a global-to-local identity mapping by a special authorization file called `grid-mapfile`. A LDAP directory stores the list of certificates corresponding to users affiliated to VOs and this information is used to create a file that contains mappings from the user identity to a local user account (real or pool). The file, distributed in all grid nodes and frequently updated, is checked whenever a host receives a request addressing a service or resource that needs authorization. Other solutions leave access control decisions to each organization composing the system. An example is the Virtual Organization Membership Service (VOMS) method (Alfieri R. et al., 2005) for central users management at VO level. The approach allows to catalogue each users on adhering on groups, roles and capabilities mapping identities with roles. Sharing web services application for a VO in a grid site offers many integration advantages, but presents security challenges: it should be only accessed by identified users according security policies locally implemented. The paper looks at the strategy of securing web services application to be shared in a specific grid environment like the Italian INFN production grid (http://grid-it.cnaf.infn.it). The case study refers to apply control access rules to java web services deployed as Web ARchive (WAR) file whose security process makes use of software developed in grid projects.

2 GRID TESTBED DESCRIPTION AND WEB SERVICES SECURITY CONCERNS

The reference grid infrastructure (http://grid-it.cnaf.infn.it) consists of several VOs representing different institutes: each one contributes with distributed grid nodes to provide resources and services. The testbed is built on a specific grid middleware (INFN-GRID) that gathers many software developed in past and present grid projects running on Linux platforms. Among these, Globus Toolkit (GT) developed by Globus Alliance (http://www.globus.org), EDG software of DataGrid project (http://lcg.web.cern.ch/LCG/), LCG middleware provided by CERN (http://eu-datagrid.web.cern.ch/eu-datagrid/) and gLite toolkit of EGEE project (http://glite.web.cern.ch) are the mainly used. They implement many grid features by means of modules installed on grid nodes. The resulting architecture (https://edms.cern.ch/document/439938/1) outlines logical machine types specifically named to represent single functionalities that VOs share in grid. Each site contributes with at least Computing Element (CE) that is a frontend node defined as a batch queue system built on a homogenous a farm of computing nodes (Worker Nodes) behind it and Storage Element (SE) that provides uniform access to storage spaces. Grid gateway allowing for command line interface to grid operations is usually the User Interface (UI) machine even if grid portal are also available. Specialized sites provide management features like the information service that keeps updated information about distributed resources and the workload management system that is responsible of match-making between best resources and job requirements and the scheduling and processing of the job itself. In this testbed both users, hosts and services must possess valid identities released by a trusted CA: the INFN institute manages a CA organized in a set Registration Authorities (RAs) to locally delegate the identification.

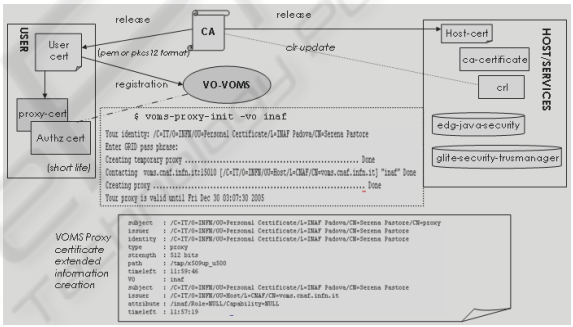

Figure 1: Distribution of grid entities certificates: creation of a VOMS proxy certificate and information attached to it.

The trusted CAs root certificates as well as the certificate revocation lists (CRLs) are available in rpm formats and installed in grid machines (in the `/etc/grid-security` directory),where credentials of grid entities that share servers or services are also stored with the proper permissions. Grid credentials are kept in a specific UI directory (the `.globus` directory) or in the browser. On demand services use delegated credentials from either the system or user. Identities are centrally managed by LDAP directory or relational database: the VO structure (each user registers to at least one VO) has brought to the VOMS client-server system for users certificates management and attributes-based authorization. Digital certificates are used for authentication, but also for single sign-on and delegation services: the proxy mechanism enable the creation, on behalf of the client, of a short term proxy certificate that will be used during a session for all subsequent

interactions with the grid. A specific proxy certificate that includes VOMS extensions is created with the command `voms-proxy-init` (figure 1). Globus proxy certificates differ from standard proposed simply adding the prefix "`CN=proxy`" to the subject DN: VOMS extensions that contain user authorization information extracted by the VOMS server are simply included in it and may be passed directly or delegated remotely to grid services which extract it for checking. Securing web services in this infrastructure is a system-wide concern: transport security (SSL) or message security (SOAP security) together with secure single sign-on play a part in providing protection against unauthorized access. Transport security realizes privacy and data integrity as well authentication of the communication end points (HTTPS), message security focuses on confidentiality (content protection) and integrity (avoiding modification). Moreover web services are deployed in specific SOAP engine that supports the idea of message handler means SOAP messages pass through a series of processing steps prior to actually delivering the message to the service implementation code. Security constraints applicable to secure web services could be addressed in grid by using the Trust Manager and Authorization Framework implemented by middleware packages named respectively `glite-security-trustmanager` and `edg-java-security`.

3 SECURING WEB SERVICES IN GRID

The approach described to provide secure web services in grid is applied to a case study of a grid application that offers facilities over distributed astronomical databases (Pastore et al., 2004) even if the security process could be generally applied. From a technical perspective a web service is a collection of related operations described by service description (XML) and network accessible through XML messaging (SOAP protocol). The application is structured as a set of java web services processed by the Axis SOAP engine and deployed in Tomcat 5 web application container running on a digitally certified CE element: access to services is enabled for authenticated grid users with specific authorization rules both by submitting a job from a UI and using graphical mode (i.e. web pages developed in JSP technology). The overall security process (figure 2) which every incoming request to web services is subjected to, can be divided into authentication and authorization steps. The client SOAP request containing the certificate is send to the HTTP listener. The authentication phase deals with the verification of the client identity through validation of X.509 certificate paths, and is performed by a gLite module known as Trust Manager.

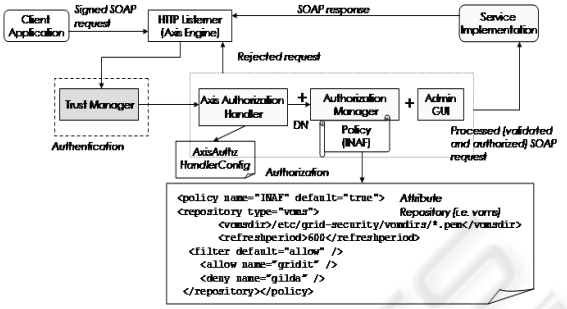

Figure 2: Modules schema for processing secure web services and description of a policy applied.

The authorization step should check the mapping between user credentials and access role and is done by an EDG module called Authorization Framework. This includes the insertion of an handler in front of the web service to be protected and the instantiation of an authorization engine (Authorization Manager) that along with an optional user management interface, it allows or denies access following the policy specified. The engine is in fact organized as a collection of Policies, one for each service that needs protection.

3.1 Trust Manager and Authorization Framework

The Trust Manager, overriding the standard Tomcat implementation, provides a java implementation of SSL protocol to support Globus proxy certificates format and CRLs of CAs. Enabling server-side-only SSL on the Web service platform, ensures messages protection in the interaction but with the limitation of securing a single communication channel. After that the client request is verified for authorization step, before passing on to the actual service implementation. The signed SOAP request is intercepted and submitted through an Axis handler implemented as a java class that protects the web service implementation. The handler examines the request, extracts the client certificate and all information embedded (i.e. subject DN and attributes), communicates with one instance of the Authorization Manager to whom delegates any decision and gets back the result of the step to access or deny the service. The policy (figure 2) applied to web service is defined in a XML configuration file (`AuthzManager.xml`) to perform a two-step mapping between the client (its DN) and a given role

and (optionally) between such attribute and a local-ID value usable by an application. These phases are specified into distinct sections: *AttributeRepository* and *TranslationMap*. Both define attribute types where to find the information needed to the authorization process. For example in the VOMS management, the `voms` type defines the use of VOMS extensions for checking rules. Otherwise the second mapping always gets the information from pluggable storage module (`AttributeMap`) that stores associations in a memory map and in turn specifies their behaviour (i.e. `regex` value allows for keys specification as regular expressions).

3.2 Practical Steps for Access Secure Web Services Implementation

The full process requires the installation and configuration of the java environment, related libraries (i.e. cryptographic libraries) and other tools (i.e. logging services) together with digital certificates for CAs and hosts involved. Client interaction with web service requires the installation of users digital credentials. Both `glite-security-trustmanager` and `edg-java-security` provide client and server java libraries to be used in applications. Trust Manager software gives configuration files and scripts to modify standard Tomcat 5 configuration and interact with it through the secure connection (8443 port) validated by host credentials. Web services deployment usually means the availability of the application as a single Web ARchive (WAR) file: the security process setup requires the settings of the modules and libraries and files inclusions before the automatic packaging made with Ant tools. Java libraries for server authentication and authorization should be available in the specific library application directory as well as the Axis ones. The configuration of the Authorization Framework requires the setup of the manager file to specify authorization policy applied, the registration of the Axis Handler with the service itself in the service deployment descriptor (`server-config.wsdd`) and the definition of handler parameters (i.e. log4j configuration file, location of the manager file, default policy, etc.) in a text file (called `AuthzHandlerConfig.txt`) whose location is specified in the web application descriptor (`web.xml`). Once deployed in the container, every client java application accesses the secure web service by using authentication and authorization libraries.

4 CONCLUSIONS

Strategy for securing web service application to be shared in grid environment as WAR file, addresses secure communication and access control. Implementation of security constraints are realized by using specific grid software. Trust Manager (by gLite) provides digital authentication supporting proxy methods. Authorization Framework (by EDG) realizes the authorization process required to access control to web services by means of a message handler technique to validating messages and an authorization engine that describing policy in XML format is compatible with the VOMS system used for VO level managing. This allows for setting specific access role for such shared resources based on VOMS attributes. Despite of software implementation, the security design is applicable for every java web services. Moreover the use of XML technologies allows to increase security by applying in future development web services security-focused specifications introduced by WS-Security, XML signatures or XML encryption standards.

REFERENCES

Foster, I. et al 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. In *Journal Supercomputer Applications*, 15(3).

The Globus Security Infrastructure (GSI) of the Globus Toolkit. At URL: http://www.globus.org

Public-Key Infrastructure (X.509). At url: http//www.ietf.org/.

Alfieri, R. et. al, 2005. From gridmap-file to VOMS: managing authorization in a Grid environment. In *Future Generation Computer System 21 549-558*.

Italian National Institute of Nuclear Physics (INFN) grid infrastructure. At url: http://grid-it.cnaf.infn.it).

LCG middleware (at url: http://lcg.web.cern.ch/LCG/) for the LHC (Large Hadron Collider) projects

European DataGrid (EDG) project (at url: http://eu-datagrid.web.cern.ch/eu-datagrid/)

gLite (at url: http://glite.web.cern.ch) software for the Enabling Grids for E-sciencE (EGEE) project (at url: URL: http://egee-intranet.web.cern.ch).

DataGrid Software Architecture Models, 2004. At URL: https://edms.cern.ch/document/439938/1

Pastore, et al., 2004. Enabling Access to astronomical databases through the grid: a case study. In Proc. *of Astronomical Data Analysis III Conference.*