Credentials Management for Authentication in a Grid-Based E-Learning Platform

Felicia Ionescu, Vlad Nae, Alexandru Gherega
University Politehnica of Bucharest
{fionescu, vnae, agherega}@tech.pub.ro

Abstract

The users of Grid-based e-learning platforms must be able to use any standard Web browser to access the e-learning and Grid resources through a Grid Portal, at any time, from any location. Standard Web security protocols, employed between a Web client and server, do not support delegation and, due to this constraint, Grid Portals are forced to give the portal special privileges to act on users’ behalf on the Grid. We solved this problem by integration of the Grid authentication mechanisms with the capabilities of Web servers using an online credential repository, called MyProxy server. Our development provides a complete and concise solution for the problem of credentials management and delegation for trusted access of users to the e-learning resources, including Grid services.

1. Introduction

E-learning refers to learning that is delivered or enabled via electronic technologies, usually as the Web over Internet network. Even e-learning systems have many advantages compared with classical learning, there are some drawbacks (such as lack of scalability and collaborative interactions) due to content delivery in Web system. These problems can be avoided using Grid infrastructure for e-learning, that offers robust, distributed collaboration and ubiquitous computing environment and secure mechanisms for resources sharing and integration as Virtual Organizations [1].

The users of Grid-based e-learning platforms must be able to use any standard Web browser to access the e-learning and Grid resources through a Grid Portal, at any time, from any location [2]. Unfortunately, there are several incompatibilities between Grid and Web security. Grid resources are generally protected by the Grid Security Infrastructure (GSI), a component of Globus Toolkit [3] that uses X.509 end entity certificates (EECs) for authentication of persistent entities and X.509 Proxy Certificates for delegation of entities’ privileges to other entities. Standard Web security protocols, employed between a Web client and server, do not support delegation and therefore Grid Portals are forced to give the portal special privileges to act on users’ behalf on the Grid.

To overcome these difficulties and obtain the secure and powerful operation of Grid Portals [4], many applications and development frameworks use an online credentials repository, called MyProxy server that enables credential delegation, single sign-on and several other facilities [5], [6], [7].

The paper describes our project that integrates Grid security mechanisms with the capabilities of Web Portals in a Grid-based e-learning platform using MyProxy server. This project provides a complete solution for the problem of credentials repository, management and delegation, offering trusted access of users to e-learning resources, including Grid services.

The solution presented here is comparable with other solutions, used in many production portals (such as National Comp. Science Alliance, NPACI [8]), and development toolkits (such as GridPort Toolkit [9]), with the advantage to be more concise and adapted to the requirements of the e-learning platform.

The paper is organized as follows: Section 2 presents a general architecture of the Grid-based e-learning system; Section 3 discusses security issues in the e-learning platform; Section 4 presents the integration of Grid and Web Portal security. Section 5 presents conclusions.

2. The architecture of e-learning platform

The architecture of the Grid-based e-learning platform (presented in Figure 1) is structured into three parts: (a) Grid Infrastructure, (b) Run-Time System, and (c) User System.
(a) The Grid Infrastructure consists of layered software components deployed in different nodes of the network. For the development of our Grid-based e-learning system we used open source Globus Toolkit 4 (GT4), which implements Web Services Resource Framework (WSRF) of the Open Grid Services Architecture (OGSA). Each GT4 node consists of containers (C, Java containers), libraries, services and different other components for publish-discovery of services, scheduling, authentication etc.

The access to the Grid services is accomplished using Java Commodity Grid (CoG) Kit [10], [11] that provides the basic APIs to the Grid to allow access to GridFTP servers, the classic GRAM services, a complete implementation of GSI and MyProxy client libraries. Java Cog Kit allows Grid users, Grid application developers, and Grid administrators to use, program, and administer Grids from a higher-level framework. Grid Infrastructure also includes MyProxy server, a component that enables Grid Portals to access Grid security mechanisms in a secure manner.

(b) Run-Time System consists of Learning Management System (LMS) and a Grid Portal. LMS coordinates all learning-related activities: stores and manages learning objects (LOs), administrates courses and tests and keeps tracks of users’ individual behavior. The LMS offers both content that makes use of Grid, as well as content that does not need Grid functionality. Grid Portal is a specialized Web Portal that provides an entry point to the LMS and Grid resources, services and data. The Grid Portal manages the identity of all users (learners, authors and administrators) and offers presentation interfaces for different learning resources (courses, presentations, tests). LMS and Grid Portal are components deployed in a Web server (Tomcat Server) developed in Java technologies (JavaServer Pages, servlets, JavaBean).

(c) The User System provides a browser-based interface for accessing e-learning platform through the Grid Portal.

This architecture combines Grid facilities with traditional Web Portal capabilities, hiding Grid complexities from user clients, that don’t need to download or install any specialized software or worry about Grid details, networks and ports.

The integration between Grid infrastructure, LMS and Grid Portal is accomplished in several aspects: Grid infrastructure offers Grid services for computational and collaboration tasks needed by learning process; available computing and storage resources are managed and shared as a Virtual Organization, using Grid publish-discovery mechanisms; security mechanisms of the Grid and LMS are integrated at the Grid Portal level. In the following sections we will present the integration of Grid and LMS security mechanisms using MyProxy credential repository.

3. Security issues in e-learning platforms

The users of e-learning platforms must be able to use any standard Web browser to access the Learning Management System and Grid resources through Grid Portals, from any location, where their Grid credentials would not normally be available to them.

These requirements need integration of Grid security mechanisms with capabilities of Web servers and browsers so that the users could to do anything through a Grid Portal that their credentials would entitle them to do. For example, a user should be able to access the Grid using a Web browser at an airport kiosk in the same manner as they could from a Web browser installed on a system on their desktop in their office.

3.1. Grid Security Infrastructure

Grid Security Infrastructure (GSI, included in Globus Toolkit) uses public key infrastructure (PKI), Secure Socket Layer (SSL) protocol, X.509 end entity certificates (EECs) for authentication of persistent entities and X.509 Proxy Certificates for delegation of entities’ privileges to another entities.

X.509 EECs provide each entity with a unique identifier (i.e., a distinguished name of the subject), the public key belonging to the subject, the identity of a
Certificate Authority (CA) that has signed the certificate, and the digital signature of the named CA.

X.509 Proxy Certificates allow an entity holding a standard X.509 public key certificate to delegate its privileges to another entity. This delegation can be performed dynamically, without the assistance of a third party, and can be limited to arbitrary subsets of the delegating entity’s privileges and for a limited (short) period of time. Once acquired, a Proxy Certificate is used by its bearer to authenticate and establish secured connections with other parties in the same manner as a normal X.509 end entity certificate, using mutual authentication protocol.

X.509 Proxy Certificates can be used to perform single sign-on and delegation over a network connection without the exchange of private keys.

The combination of a public key certificate and the matching private key is named credential. Usually, the private keys associated with X.509 public key certificates are protected either by encrypting them with a pass phrase (if stored on disk) or by requiring a PIN for access (if stored on a smart card). While this method serves to provide a high protection of the private key, it can be prohibitively burdensome if the user needs to access the private key frequently for authentication to other parties.

Proxy credentials solve this problem by enabling single sign-on: that is, allowing the user to manually authenticate once in order to create a proxy credential which can be used repeatedly to authenticate for some period of time without compromising the protection on the user’s long-term private key. This is accomplished by creating a new key pair (composed of a public and private key), and by subsequently using the user’s long-term private key to create a short-lived Proxy Certificate. The Proxy Certificate binds the new public key to a new name and delegates some or all of the user’s privileges to the new name.

The Proxy Certificate and the new private key (that form together a proxy credential) are then used by the bearer to authenticate to other parties. Since the Proxy Certificate has a short lifetime, it is typically permissible to protect it in a less secure manner than the long-term private key. In practice this means the proxy credential private key is stored on a local file system and is protected by only local file system permissions, which allows the user’s applications to access it without any manual intervention by the user.

X.509 Proxy Certificates can also be used to delegate privileges from an issuer to another party over a network connection without the exchange of private keys. This delegation process requires that the network connection be integrity-protected to prevent malicious parties from tampering with messages, but does not require encryption as no sensitive information is exchanged.

3.2. Web Portals security

Web Portals do not integrate cleanly with existing Grid security systems such as the Grid Security Infrastructure (GSI), due to necessity to keep private key of the Grid credential protected by a password (pass phrase) and the lack of delegation capabilities in Web security mechanisms.

Since Grid credentials are typically stored as files on a file system and the private key must be kept private, a user must have secure access to the file system in order to use his credentials. This means that the users are unable to use them when they are away from the computer where their Grid credentials are located. While smart cards and other hardware tokens (which can store credentials on a portable medium) would be a potential solution to this problem, support for these devices has not been widely deployed.

The second problem in accessing the e-learning Grid Portal is the lack of delegation capabilities of Web browsers (used as clients with Web-based Grid Portal). Web browsers can use Grid credentials for authentication (even the vast majority of Web authentication is accomplished with username and password), but lack the GSI’s ability to do credential delegation. This means that, although the users can authenticate to the portal, they cannot delegate authority to the portal to act on their behalf, meaning the portal is severely restricted in usefulness.

We solved these problems using MyProxy credentials repository system and several modules included in Grid Portal, which assure the integration of Grid and LMS security mechanisms by authentication in the Grid Portal. MyProxy is an X.509 credential management system that can properly handle both X.509 end entity credentials and X.509 proxy credentials encrypted by user-chosen pass phrases. It consists of a repository server and a set of client tools that can be used to delegate (store) to and retrieve credentials from the repository.

4. Integration of Grid and Web security

Authentication of users in the Grid Portal of the e-learning platform consists of two steps: authentication in the Web server and authentication in the Grid. Authentication in the Web server is implemented with a JavaBean class that is instantiated in the login JSP page of the portal. This bean receives username and password and checks them against the users’
information stored in a database; if succeeded, Grid authentication of the user is checked.

For authentication in the Grid (for access Grid services and resources) a user must provide a X.509 proxy credential that can be obtained using MyProxy repository.

### 4.1. Credential delegation in MyProxy server

In order to access the Grid services and resources of the e-learning platform, a user must have a Grid credential and uses this credential to delegate his privileges to the Grid Portal.

To delegate (store) a proxy credential in the MyProxy server, the user would run myproxy-init client program (contained in the MyProxy server package) on the machine (or logged in a secure manner - e.g. an encrypted Secure Shell session) where he can access his X.509 credential (using the passphrase that encrypts the private key associated with this credential) and delegate a proxy credential to the MyProxy server repository along with authentication information (pass phrase) on that server.

Figure 2 shows the steps executed for the delegation of privileges by creation of a proxy credential over a network connection to MyProxy server.

In step (4), myproxy-init client and MyProxy server perform mutual authentication and establish an integrity protected channel (using the SSL protocol). Afterwards, the myproxy-init client asks the MyProxy server to request delegation for a subset of user’s privileges. For this, the MyProxy server generates a new public and private key pair and a proxy certificate request with the new public key (step 5) and sends proxy certificate request back over the secured channel to the myproxy-init client, to be signed. (step 6)

In step (7), myproxy-init client uses the private key associated with user’s certificate to sign the certificate request, generates a new proxy certificate containing the newly generated public key and sends it back over the secured channel to the MyProxy server. Afterwards, myproxy-init client ask a passphrase and a confirmation from the user (steps 8 and 9) and sends it to MyProxy server to be used to encrypt the private key of the proxy credential (steps 10).

Finally, MyProxy server stores a new credential, consisting of the new proxy certificate signed by the myproxy-init client and new private key, encrypted with the passphrase received from the user (step 11).

To execute these operations the user must have a Grid credential stored on host A and access this host in a secure manner. After the user have delegated the proxy credential in the MyProxy server, he can use this delegation from any place (host) in the network, without need to store (multiply) the long-time credential on other hosts, without the need to transmit over the network the private key or the pass phrase that encrypts private key of his long-time Grid credential.

### 4.2. Credential retrieval from MyProxy server

To obtain a proxy credential from the MyProxy repository, the user must provide a username (that can be the same as portal username) and the repository passphrase (the passphrase provided by the user when he has delegated a proxy credential in the MyProxy server). Figure 3 shows the steps executed by different software components for proxy credential retrieving from the MyProxy server.

In our e-learning Grid Portal, proxy credential retrieving is implemented with a module (package org.globus.portal.proxy) that contains a JavaBean class (MyProxyBean class), which is instantiated from the JSP login page of the portal. This class offers the public function establishContext() that retrieves users’ proxy credential from the MyProxy server and stores it in an established location (/tmp/x509up_uid, where uid is the user identifier).
This bean uses Java Commodity Grid (CoG) Kit (package org.globus.myproxy) in order to connect to the MyProxy server. A short fragment from MyProxyBean.java file is presented in Figure 4.

```java
package org.globus.portal.proxy;
import org.globus.myproxy.MyProxy;
public class MyProxyBean{
    public boolean establishContext(String u, String p){
        int result = getProxyX509(u, p);
        if (result == 0) return true; else return false;
    }

    public int getProxyX509(String u, String p) {
        GlobusGSSCredentialImpl portalProxy = null;
        ExtendedGSSCredential userProxy = null;
        // Generate Host Certificate
        GlobusCredential hostCert = new GlobusCredential(...);
        portalProxy = new GlobusGSSCredentialImpl(...);
        // Connect to MyProxy server
        MyProxy myProxy = new MyProxy(...);
        // Obtain user credential
        userProxy = (ExtendedGSSCredential)myProxy.get(...);
    }
}
```

Figure 3. Credential retrieval from MyProxy server

The steps executed by different software components for proxy credential retrieval from the MyProxy server are explained below.

Initially, the user logs in the Web Portal sending required information (username and password) from a Web Browser (step 1) and the portal checks this information against the values stored in the database (step 2). If user is authenticated in the portal, the login module asks user’s passphrase for proxy credential in the MyProxy server (steps 3 and 4).

In step 5, the login module calls establishContext() method on the MyProxyBean object. In this method, MyProxyBean finds the proxy credential of the portal, instantiates a MyProxy client (as instance of the org.globus.myproxy.MyProxy class from Java CoG Kit) and calls get() method on this object (step 6).

In next step (7), MyProxy client generates a public and private key pair and an unsigned proxy certificate with the new created public key and establish a secured channel with MyProxy server using their own credentials.

The unsigned proxy certificate is sent by MyProxy client to MyProxy server to be signed (step 8) and MyProxy server signs and sends back the proxy certificate to the MyProxy client (step 9).
This user proxy credential (signed proxy certificate and associated private key) is then returned by MyProxy client to the MyProxyBean (step 10) and MyProxyBean stores it in an established location in local file system (step 11).

After obtaining the user proxy credential, the Grid Portal securely accesses the Grid services in behalf of that user (steps 12 and 13).

The operation of logging out of the portal deletes the user’s delegated credential on the portal. If a user forgets to log off, than the credential will expire at the lifetime specified when requested from the MyProxy server that is normally on the order of a few hours.

This process could then be repeated as many times as the user desires until the credentials held by the MyProxy repository expire, at which point the user would need to rerun the myproxy-init program from a location where his permanent credentials were available and delegate a new set of credentials to the repository.

5. Conclusions

In this paper we described our design and implementation of an integrated authentication infrastructure for a Grid Portal that combines username and password authentication of users in the portal with Grid credential delegation using MyProxy server. In this way, the users don’t need to re-authenticate while access Grid services and e-learning resources from any host in the network and passwords used for proxy credentials generation and retrieval by the portal are never stored and travel the internet via a secure https connection.

This approach can be adopted for any Grid platform accessed through a Grid Portal, and allows the protection of Grid resources by proxy credential delegation based on Grid Security Infrastructure (GSI).

Acknowledgment

This work was developed under Romanian Grant CNCSIS code 44 (UPB part 14) started in 2006.

References


