Integration of authorization for grid

B. Radic, E. Imamagic, D. Dobrenic
University Computing Centre, University of Zagreb, Croatia
{branimir.radic, emir.imamagic, dobrisa.dobrenic}@srce.hr

Abstract. Today there are numerous authorization systems both for protecting and managing individual and distributed computer systems. Systems for authentication and authorization for distributed systems work only on software they were developed for. A complete system of authentication and authorization that works with different grid middleware is not yet developed. What can be done is to either implement a solution which is most adequate for a given distributed system and its purposes or try to make an adequate solution by combining multiple systems.

In this paper we present problems and challenges when integrating authorization solutions of different grid systems. We present our experiences in enabling different systems to work together and describe extensions which we made in order to enable multiple authorization systems to work together.

Keywords. Grid, authorization.

1. Introduction

Grid systems comprise of a number of components that interoperate to create a complete heterogeneous distributed system. Authorization requirements in general are divided into generic and grid specific. Each grid based on its’ purpose and goal has specific demands and specific infrastructure. Number of components and the way in which components interoperate is also different in each system. Decision, which systems to use for any required functionalities is based on the purpose of given grid. Certain functionalities don’t need to be implemented. We can use an example of a grid that has users who all have different roles and different access rights for accessing resources of the grid. In this example a need for a system for access control exists. But it is also important to notice that it is completely unnecessary to have an access control system based on group management.

This type of a decisions have to be made whenever implementing and managing security mechanism on any grid.

Goal of this paper is to describe in detail our experience with different grid systems, for achieving authorization, access control and auditing. We have developed certain extensions to ease user authorization on multiple middleware systems in CRO-GRID [2] and we will explain how they manage users. Further more we will explain for any system mentioned its’ main purposes, its’ advantages as well as disadvantages.

2. Grid security problem

Grid system consists of subsystems spanning different administrative domains. These domains can all have their own security policies and rules. A grid authorization system must be implemented in such a way that all those policies and rules are respected. Demands in grid systems also include scalability, remote delegation and distributed administration of both users and resources.

Grid systems commonly have a large number of users. These users mostly belong to certain organizations which contribute compute and/or other resource types to the grid. Users’ rights of access can be regulated by a number of factors. First, every user that is a member of a certain organization possesses the rights to access resources based on his role in that organization. Every user can also be a member of multiple organizations and/or have multiple roles in one organization. Users also don’t have to be members of any organization.

Managing all this data is demanding especially as one of the tasks of the authentication mechanism is to enable a single sign on. Single sign on means that user has to authenticate only once to the grid security system and he will automatically have all the rights that are assigned to him on all the resources for the entire duration of that session.
If all the data about user access rights would exist on every accessible grid site then all data would have to be up to date and, as a high number of copies exist, a large overhead would be created. On the other hand keeping all the data about user rights on a single site creates issues of availability as well as creates a large overhead in network communication to that site and security risk.

There are two approaches [1] to administering users that are most commonly used.

First common approach is managing users individually. This means that every user has a profile where all the data about his access rights is stored. In this approach adding or taking away access rights is simple. However enforcing the access rights is much more demanding. First problem is how to implement communication during authorization. One idea is to automatically transfer all the data about a user to the site where user logs on. Problems here are overhead and availability. Overhead is created because complete set of the data about a user rights can be large and the availability issue is that accessing any data on a site is not possible if the central site with user access data isn’t accessible. Second idea is to do a check of the users rights every time a demand for data or resource access is made. In this solution the problem of availability also exists and is even more threatening then in the first solution as central site for security must be available for the entire time of the users’ activities on the grid. Second problem is that site with user access data can become a bottleneck and users might be forced to wait for their demands to be processed.

Second approach is to create levels of security. In this approach every user gets his level of security clearance based on his rights of access. Also every available data or resource that is accessible gets additionally described by a level of security needed to access that resource. User can access a resource if his clearance level is equal or greater then the clearance level assigned to the resource he is trying to access. This solution is aimed at systems that in time change little or not at all. Grid, being a constantly changeable environment, is not suitable for this type of a security policy implementation. Implementing this solution would require making a separate policy for every grid site. Level that is assigned to every user would be different on every site. Also there would not be an efficient mechanism for giving very specific rights to users. If a user needs to be given an access to just a small number of files he would have to be given the right to access a whole class of files that belong to that authorization level. Alternative is to create a distinct level which will enable the user to access that files. This approach in that case becomes just a more complicated version of an approach that doesn’t use levels of security.

These two ideas are the basics for ensuring users’ authorization of data and resources in common systems. Because of the problems, which we described earlier in the text, our opinion is that authorization should be handled in a different way. In the next chapter we will describe in theory how the problem can be solved and in the chapter after we will describe practical solutions that we use and our experience with them.

3. Theoretical view of grid authorization solutions

Because of the number of sites with data and/or resources that need to be managed with an authorization mechanisms and because of the number of users using those resources an authorization schema is needed.

Main concept used to identify users and ensure the security of grid systems are certificates. Although their main purpose is for authentication, their role in authorization is also important as access is allowed or disallowed based on them.

Certificates are used for authenticating users and services. To fulfil its role a certificate must have certain components.

A minimal certificate consists of a subject name, public key of the certificate owner, identity of the Certificate Authority (CA) and a digital signature of the CA. Subject names who is the person or the service that the certificate represents. Public key is used to prove identity provided by the subject. Digital signature of the CA is used to prove that the CA named in the CA identity truly did issue that certificate.

Certificates are in a grid environment used for authentication and creating a temporary version of them – proxy certificates. These proxy certificates are in most aspect identical to the certificates used to issue them except the fact that they are valid only for certain, usually short, period of time. Proxy certificates can also be used to transfer only a subset of user privileges. A way that the certificate is used on grid varies
in different systems, but as a rule the grid security systems use certificates.

There are currently three main practical approaches to managing security in grids. First approach is a direct implementation of the approach where users are managed individually, and the third approach is a bit modified version of that approach. The second approach is for grid modified version of the security level approach.

3.1. Individual account creation

First approach is to create individual accounts for each grid user [4]. On every site a mechanism for defining which user can access that site exists. For accessing a site user must prove his identity. This is typically achieved by using certificates. In first step user uses username and password to prove his identity anywhere on the grid. A proxy certificate is issued and is given to the person or a service that demanded it. Based on that issued proxy certificate user can identify to all the sites for as long as the proxy certificate is valid. This enables single sign on. Each site however can decide not to allow access to the user. Access is granted only if all the components that make certificate are approved by the local authority. User named in the certificate and his identity must be verified through his public key. Certificate Authority noted in the certificate must be trusted by the site and the digital signature must be successfully verified by the site.

Two main problems exist in this schema that are both connected with the way in which user identity is managed. When every single user must exist on every grid site there is a problem of identity mapping from grid identities to local identities. Same local identity might exist on different sites. These identities must be mapped into different global identities. Also a mechanism for identity mapping must resolve conflict when a global identity is mapped to a local identity on a site where already exists a local identity with that username. Second problem is user management on a global scale. Adding a user to grid also requires creating an account for that user on all the sites that he has access to. Although a simple revocation of a certificate should prevent an existing user whose privileges on the grid are removed from accessing the grid, completely removing the user requires changing all the sites that he had access to.

Main advantages of this approach originate from the same fact from which originate the disadvantages and that is the usage of local mechanisms. Namely, once user is (through the usage of proxy certificate) authorized, and his global identity is mapped to a local identity, then all management can be done in the same way that it is done with any other system users, ignoring the fact that this user is actually a grid user. This eventually results in the simplicity of user auditing and logging. After login grid user is automatically capable of using protocols native to the system, like any user of the system.

3.2. Virtual organization management

Second approach is creating user groups based on their access rights and management privileges. Such groups are called virtual organizations [12] (VO). This approach is very useful when managing multiple users with identical access rights to grid resources. Benefit of virtual organizations is simplification of management of user groups and resource groups. Usage of VOs also increases security in general. This approach uses a modified idea of security levels. Instead of giving the access rights and management privileges to individuals, such rights are given to virtual entities – virtual organizations. Every such organization is given an administrator. Members of the virtual organization are added, removed and managed by the administrator of the VO. Resource managers give access rights to a certain VO by authorizing the administrator of that VO. Administrator can then manage access rights to that resource. In most cases there is no hierarchy between VOs. When using grid resources a user is not identified as the user himself but rather just based on his role(s) in organization(s). In this approach it is simple to manage large numbers of users.

Disadvantage of this approach is that users can’t simply use services that are common part of the resources operating system. All access to the services that users require must be given to them through the VO. Second problem is auditing. All the tasks are logged not as taken by the user but rather as taken by the VO, only in some implementations as taken by the VO member. When a problem of any sort occurs, it is a demanding task to detect which user caused problems.
3.3. Virtual account management

Third approach [6] is to create a new user on a local site every time a grid user accesses that site. Grid user is then mapped to the newly created local user account and works with the privileges which are given to that account.

Advantage of this approach is flexibility and the fact that the same account is never used for more than one grid identity.

Disadvantage is that this method interferes with local account management. There is also a theoretical chance of reaching the system's maximal account number. Finally, control of account creation, configuration and termination is made hard by this approach.

4. Grid authorization solutions in practice

In this section we will in short explain the authorization system of the CRO-GRID and present the practical decision we are facing.

We are working on the CRO-GRID Infrastructure [3] project. Goal of the project is to create a grid infrastructure that will be used by different scientists in Croatia which require significant compute power for their research.

This grid is not specialized in the genre of calculations that will be done on it. Also the type of the research that it is used for will change over time. With these demands we decided to implement several grid middleware technologies in parallel so we would be capable of fulfilling as much different user demands as possible. Technologies implemented are as follows: Globus Toolkit 2.4 [5], Globus Toolkit 4 and Uniform Interface to Computing Resources (UNICORE) [11].

Globus Toolkit (in future reference Globus) is a de facto standard for building grid middleware. The version 2.4 of Globus was the last version which is not based on proprietar protocols. It has services for monitoring, job execution, data transfer and some basic security mechanisms. This version is (in spite of developing Open Grid Service Architecture (OGSA) services, [8] which it doesn’t support) still very much in use.

In May 2005 a newer version of Globus was released. This version has brought new services as well as Web Services Resource Framework (WSRF) [7] compliant versions of services present in version 2.4. Reason that we installed both versions of Globus is to have a stable version of core services through the usage of version 2.4 and to have WSRF compliant versions of those services, which are not so thoroughly tested but are more advanced and offer additional functionalities.

UNICORE is a middleware system implemented in Java. Advantages of UNICORE are simplicity of installation and configuration, portability, integration with different cluster and supercomputing systems for data management, and finally an advanced graphical interface. To access UNICORE user must have its’ graphical interface installed on local machine.

Both Globus 2.4 and Globus 4 use X.509 certificates [10] in PEM format for authentication of both users and services. In order for the user to be allowed to access a resource there must be a record of the user in an authorization file. That authorization file in Globus systems is grid-mapfile. Entries in grid-mapfile consist of two parts. First part is the subject name of a certificate and the second part is the username of the local user. When a certificate is used for authorization there are two possible outcomes. If a subject of the certificate does not exist in the grid-mapfile then user has no rights of access to the resource and his request is denied. If on the other hand there is an entry with the subject provided by the certificate then user that is trying to access resource is the one to which other part of grid-mapfile entry refers to. User in that case can be authorized. He is given rights of the local user that the global identity is mapped to.

UNICORE uses a keystore for keeping user and CA certificates. Format used for work with user certificates in UNICORE is PKCS12 [9]. In this format both user certificate and user key file are placed in one file. CA certificates are stored in PEM format. An adequate user certificate and CA certificate must be stored on every site from which user wants to access grid. To authorize users, UNICORE uses a UNICORE User Data Base (UUDB). Through that database user certificates are mapped to local user names that are used on target system for every action that user takes. Each certificate can map only to one local user name, but multiple certificates can map to a single user name. This enables creating just one local user for usage by multiple users with identical access rights. Main component of UNICORE, the Network Job Supervisor (NJS), can be configured to accept proxy certificates and end certificates. End certificates are the certificates issued by the CA. When working this
way, upon receiving an authorization request, NJS first tries to match an incoming certificate with an UUDB entry, if that fails it will try to match the signer of the certificate with a UUDB entry. If any of the attempts successfully returns a match then NJS allows the user to take actions based on his rights.

5. Our approach to authorization

Certain ways exist to use the same certificates for authentication on different grid middleware, but there is no unique solution for authorization on them. That is why we decided to use multiple authorization mechanisms in parallel, enabling users to access resources through any of them. This is a variation of the approach described in 3.1. The complete procedure of obtaining a grid account is shown on Figure 1.

![Figure 1. Procedure of obtaining a grid certificate](image)

When a user requires access to CRO-GRID resources he must make a demand for issuing a certificate. Certificate demand can be made anywhere on the grid, and a policy is implemented on every site which forbids issuing an invalid form of a certificate request. A valid certificate request needs to be sent to the grid administrator. Administrator checks validity of the request and checks that the username in the certificate is unique to the grid. This check needs to be made so it would be possible to accept requests which aren’t made on our grid sites. Valid requests are then forwarded to the CA. Based on a valid certificate demand CA issues a user certificate. Certificate is not sent to the user, but rather back to the grid administrator. Grid administrator then uses our software on the certificate to create a grid user account. Grid user account is automatically created for both Globus systems and UNICORE system simultaneously. User account is created on all CRO-GRID sites, and in the same time user certificate and user private key are stored in directories as is later needed for users’ authentication. User data is also simultaneously added to grid-mapfile of Globus and to UNICORE UUDB. Once this is completed user is informed that his grid account is created and is ready for use. For each user a random password is created this password which the user must change. All the data is passed to user in a secure way. This procedure is done only once so the overhead of the user certificate transfer is insignificant.

![Figure 2. Accessing grid resources](image)

User can then start using the grid resources. As can be seen on Figure 2 first thing that any user does is to authenticate. Authorization system verifies that there is a record representing him in the user database. This step is common for both Globus and UNICORE users. Once user is found in the user database his request is processed by the appropriate authorization mechanism. Requests made to the Globus system are checked to the grid-mapfile and the result of the check is returned, while validity of the requests made to UNICORE is checked in the UUDB.

Authenticated users are then authenticated to use grid resources. The way that user access and use their certificates is left to the users themselves. For work on Globus systems a proxy certificate is needed. Users can log on to any grid site and use the certificate on that site to issue a proxy certificate or use some certificate repository to store a certificate and connect to that repository when they wish to issue a proxy certificate. In the same way a PKCS12 certificate for usage on UNICORE is freely managed by the user.

This approach offers a simple and reliable method of auditing as every action is logged
automatically in the same way like for every other system user. Problem is managing larger number of users where this procedure must be repeated for every user even if they have identical access rights. Modifying access rights can be done by editing the access rights of the user on the site. Modifying access rights of a user on the entire grid requires that changes be made on each site.

Removing authorization rights of users can be done in several ways. Removing access rights to an individual site from a user can be done by simply removing entries in grid-mapfile and UUDB or by deleting user account on that site or to be on the safe side doing both. Doing any of the two prevents user from accessing the site and from using resources.

Fastest way to completely remove access rights from a user is to revoke his certificate. On every site there is a Certificate Revocation List (CRL) file in which is a list of all the revoked certificates. Every certificate that is in that file is revoked and every demand made with that certificate is denied.

To completely remove a user it would be necessary to remove all his files and accounts from all the grid resources. At this point only way to do that is manually on every site.

6. Conclusion

Grid authorization is a complex problem because of the complexity of the system and its dynamical nature. This is very present for the CRO-GRID as our grid is unspecialized and usage demands can change based on the needs of users. Present solution simplifies all the procedures that are currently needed.

The system we created simplifies user creation while keeping the possibility of using multiple grid middleware systems. More precisely through the usage of this system it is possible to simply use multiple authorization mechanisms and only one mechanism of authentication.

By improving features of this implementation it will be possible to completely manage users and their privileges and keep multiple grid middleware operational.

7. Future work

As mentioned earlier we predict certain changes in the behavior of CRO-GRID, and our future work, just like our current work, must adapt to those demands. First step will be expanding the security authorization system with mechanisms for automatic removal of users and revocation of the certificates. Second phase will be testing and implementing other solutions, primary the ones which include management of groups of users and modifying those solutions to fulfill CRO-GRID needs.

As this is a young project there was hardly any need to revoke user accounts or rights. With time administering CRO-GRID will demand less and less account creating and more management of existing accounts like removing or limiting access rights of existing users.

8. Acknowledgements

All given results were achieved through the work on the TEST program – Technological research-development projects with the support of the Ministry of science, education and sports.

9. Reference