ROLE BASED ACCESS CONTROL FOR GRID ENVIRONMENT USING GRIDSIM

Simarjit Kaur¹, Sukhjit Singh²

ABSTRACT
Security is a major issue in highly computational resource sharing environment [5]. Computer-based access controls can prescribe not only who or what process may have access to a specific system resource, but also the type of access that is permitted. Access control is a security measure that protects shared resources against unauthorized accesses. The resources that are protected by access control are usually called objects, whereas the entities whose accesses are regulated are termed users (or subjects). Access control [6] is required to preserve and protect the confidentiality of the information of the system and its resources. Access control is implemented by using simulated grid environment.

KEYWORDS:--RBAC, DAC, MAC, XML, Gridsim.

INTRODUCTION
Large-scale distributed computing environments, or computational grids as they are sometimes termed, couple computers, storage systems, and other devices to enable advanced applications such as distributed supercomputing, computer-enhanced instruments, and distributed data mining. While scalability, performance and heterogeneity are desirable goals for any distributed system, the characteristics of computational grids lead to security problems that are not addressed by existing security technologies for distributed systems. Access control is a security technology that is applied extensively to preventing unauthorized access to information and system resources. Many models have been developed and studied to construct and manage access control systems. The Role Based Access Control (RBAC) Model is used widely in distributed network systems to protect shared data and information. It uses security specifications to control the access to critical resources and avoid security violations. However, the security specifications are often defined as an ad-hoc manner in the deployment stage after the software design. This causes increasing vulnerabilities to the IT systems. Moreover, the security specifications are normally documented in files. Hence, a tool to support the security specifications editing and generation can increase the productivity and the quality of the security specifications. Role Based Access Control (RBAC) is way to restrict the unauthorized access of resource in grid environment. The proposed access control tool uses roles as basic entity for authorization. In Grid environment demands solutions to support security policies and management of credentials. Support for remote access to
computing and data resources is to be provided. Access control to services needs to be provided via robust security protocols and comprehensive security policies. In the proposed access control tool access control policies and user roles are defined in XML document. To provide grid environment simulation is done by using Gridsim toolkit. Gridsim is a toolkit for modelling and simulation of distributed resource management and scheduling for grid computing. GridSim[12] is a software platform that enables users to model and simulate the characteristics of Grid resources and networks with different configurations. Study Grids, or test new algorithms and strategies in a controlled environment. By using GridSim, the users are able to perform repeatable experiments and studies that are not possible in a real dynamic Grid environment. Some of the GridSim features are given below:

- It allows modelling of different resource characteristics and their failure properties;
- It enables simulation of workload traces taken from real supercomputers;
- It allocates incoming jobs based on space-or time-shared mode;
- It supports reservation-based or auction mechanisms for resource allocation;
- It has the ability to schedule compute- and/or data-intensive jobs;
- It provides clear and well-defined interfaces for implementing different resource allocation algorithms;
- It allows modelling of several regional Grid Information Service (GIS) components.

![Fig. 1 Architecture for Gridsim platform and its components](image)
METHODOLOGY
To implement access control proposed framework various access control techniques are studied. The proposed access control tool is based upon roles. Different roles have different access rights. The role is assigned to a person before accessing the resources of the computer. In the proposed work there are different types of roles defined in XML file. Depending on the responsibility different access rights are given to different roles based upon priority in the organization. The user of particular role type access the resources which are restricted by access rights imposed upon that role. for the access control tool to work. These are Netbeans IDE which is free open source software and Gridsim toolkit which can also be downloaded easily. GridSim has employed a layered and modular architecture for Grid simulation to leverage existing technologies and manage them as separate components. A multilayer abstraction and architecture for development of Gridsim platform and its applications is shown in Fig. 1. Gridsim creates gridlets and resources on each gridlets to use as simulated grid environment. The GridSim [11] toolkit provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers. The proposed tool is implemented on the top of Gridsim toolkit. For that first the toolkit is installed on the computer. Then Gridsim package is imported to simulate Gridsim environment using netbeans IDE. The tool is further simulated in Gridsim. The users can be created in Gridsim and there can be more than one user at a time using the grid. The users are assigned roles according to that the users can use the resources for job execution. The having highest priority can use resources on high priority basis than the low priority user in the grid environment. Below are the steps to be followed in JAVA code to create a Grid Resource.

- Create an object of type MachineList to store one or more Machines

  \[\text{MachineList mList} = \text{new MachineList();}\]

- A Machine contains one or more PEs/CPUs. So we create an object of type PEList to store this PEs before creating a Machine.

  \[\text{PEList peList1} = \text{new PEList();}\]

- Create PEs and add these into the object of PEList created in step 2. We have to specify the unique ID of the PE as first parameter and its MIPS (Millions of Instruction per Second) rating as second parameter.

  \[\text{peList1.add(new PE(0, 377));}\]

- Create a Machine with its unique ID and the PEList associated with it.

  \[\text{mList.add(new Machine(0, peList1));}\]

- Repeat the steps from step 2 to step 4 to create additional number of machines.

- Create a Resource Characteristics object which will store the properties of a Grid Resource.

  \[\text{ResourceCharacteristics resConfig} = \text{new ResourceCharacteristics(arch, osmList, ResourceCharacteristics.TIME_SHARED, timeZone, cost);}\]
In the final step we create an object of Grid Resource specifying its name, communication speed, peak load, off-peak load, holiday load and list of holidays along with the Resource Characteristics object which was created in the step 6. Resource creation is shown in Fig. 2.

The steps to create Grid User(s) in GridSim are straightforward, but each User must have a unique ID. The steps to create a Grid User are as follows:

   ```java
   ResourceUserList userList = new ResourceUserList();
   ```

2. Add to this list Grid users specifying a unique ID, first user has to have ID equal to 0. Just keep on adding Grid users to this list if we wish to create more of them.
   ```java
   userList.add(0);
   ```

In the terminology of GridSim, a job which can run sequentially and independently on a Grid Resource is called a **Gridlet**. After the creation of Grid Resource we create Gridlets in the GridSim which can then be submitted to the latter. We need to specify the length of Gridlet, its output file size, its input file size, its unique ID for simulation. Gridlet creation can be done in two modes first is the manual option and second is with the use GridSim Random functions to take care of the statistical needs of highly unpredictable Grid environment simulation.

The steps to create Gridlet(s) in manual modes are:

1. Create an object of type GridletList
   ```java
   GridletList list = new GridletList();
   ```

2. Create an object of type Gridlet specifying its unique id, length, input file, output file size in types integer, double, long integer and long integer respectively.
Gridlet gridlet1 = new Gridlet(id, length, file_size, output_size);
3. Add the object created in step 2 to the Gridlet list created in step 1.
list.add(gridlet1);
4. To create more Gridlets repeat from step 1.

After creating the grid user, resources and gridlets for the proposed tool to start simulation. The user will login to the access control tool by giving the credentials. The authentication module will check the credential by using xmlroleparser class designed in Java. Then the user will proceed with the tool and start accessing the resources.

DISCUSSION
The proposed work is platform independent and data storage is also very flexible using XML can be run on any computer without any dependency. The proposed framework is based upon Role-based access control seems to be a better choice for authorization than MAC since it allows XML documents to be stored in XML Store by subjects each controlling their own level of access to other subjects.

Since MAC does not allow subjects to entirely determine its own access control for stored XML documents. With DAC, document owners need to define access policies for stored resources but does not put any restrictions on XML. RBAC model solved the issues of MAC and DAC both by providing access control based upon roles. There are different types of roles defined in the access control tool. There are priorities defined grid environment.

![Fig. 3 Resource Assignment to various users by using Gridsim](image)

The users are assigned over roles. The role with highest priority can acquire all the demanded resource for job execution in simulated Roles when created. The user starts the access control in simulated environment by giving the credentials. The credentials are checked by
xmlRoleParser class for access rights assigned to that user role. At the time of authentication user is compared with the role by xml parser class of Java. If the role and user match in XML document then the server starts and ready to communicate.

Gridsim toolkit is a platform independent and the access control tool is implemented by using to create simulation. The proposed tool is a Java based tool which provides access control based on role. It also provides resource sharing in a restricted way so as to provide secure resource sharing.

To provide secure resource sharing roles are defined and policies are implemented by security manager in Java. The resources are assigned to various users to execute the job at corresponding Gridlets. The cost job execution is shown in Fig. 3.

The main contribution of this proposed work is the design of access control tool for grid and distributed environment. The specific contributions are:

a) It is providing better role management by assigning priorities to different role in the system.

b) Platform independence is provided by this tool.

c) It is providing access control to end-user, to improve the usability of system resource without compromising the confidentiality requirements.

d) It is an access control model based tool that incorporates various roles and supporting various security policies.

The roles are stored in XML document. Each user has assigned role under which user’s credentials are stored such as user name and password which are checked at the time of authentication.

CONCLUSION

This proposed work implements a new a tool for providing resources to various jobs which are executed in grid environment. New proposed tool implements access control to various resources based on roles. Roles reading and parsing have been presented. The proposed tool maps the user to roles defined in the access control system. Each user is assigned to a different role according the access rights given to the user. Each user’s role is read from the XML files and then access rights are assigned. During the working of the access control tool distinction between different user’s roles is shown. Also the proposed role mapping architecture makes it possible for two nodes to interact and authorize users for resource access. The access to resource is controlled by giving access to only user of defined roles by just comparing the user with XML file. The proposed work provides convenient role evaluating and decision making approach. It also provides interface to users, administrator.

The simulated Grid environment is created as test bed for the new developed tool. Due to the difficulty in creating a real Grid environment
Grid Simulator is used. The simulator which is used for simulation of environment is GridSim which is java based toolkit can be installed on any operating system without any dependency. The proposed work is implemented above the toolkit in Java using Netbeans IDE (open source software).

The authorization policy in the proposed work makes it more efficient for access control in grid and networks. Due to priority rule approach it becomes easy to authenticate only user role is compared with the roles defined in the framework. Further the access to various resources can depends upon the role with user has logged into the access control tool. The access will deny if user does not match with role type defined.

The proposed work for access to resources is based on resource allocation in GridSim provides more optimal and feasible way to allocate resources. The resources that are adaptable to the requirement of the job with less cost are selected to run the jobs.

REFERENCES