Scheduling Working Group
Scheduling Working Draft: 6.1
Category: Informational

Keith R. Jackson
Lawrence Berkeley National Laboratory
Volker Sander
Forschungszentrum Jülich GmbH

September 2000

Security Requirements of the Scheduling Working Group

Status of this Draft

This draft provides information for the grid scheduling community. Discussions and suggestions are requested. Distribution of this memo is unlimited.

Copyright Notice

1. Introduction

Computational Grid schedulers may be used to schedule access to a wide variety of resources existing in many different security domains. These schedulers must be able to identify, authenticate, and authorize users according to local policy before attempting to schedule any resources. Users must be able to identify, and authenticate the scheduler they connect to. These schedulers will need to be able to securely act on behalf of the user as they attempt to schedule resources on her behalf. The user should be able to delegate to the scheduler some sub−set of her rights, including possibly the right to further delegate those rights.

This document attempts to capture through a series of usage scenarios the security requirements and implications of grid scheduling. It is not intended to provide an exhaustive list of all instances where security operations may be applied, but instead attempts to enumerate the most common places where security operations are required. Due to the relative importance of delegation to Grid scheduling, a separate section lists some of the important delegation requirements.

2. Usage Scenarios

2.1. Definitions

In the following usage scenarios the term **user** will refer to the person attempting to access the resource. The term **principal** will refer to any identifiable entity, either human or computer, that wishes to access resources. A **security gateway** is a process that mediates access to a resource and performs the necessary security operations, e.g., the Globus gatekeeper. The term **credential** refers to a set of attributes attesting to, or establishing, the identity of an entity. **Impersonation** refers to the ability for one principal to allow another principal to act as if it were the original entity. An impersonation credential does not contain any information about the identity of the entity the rights were delegated to. **Delegation** is similar to impersonation, except that the delegated credential contains the identity of the principal the rights are delegated to. **Limited delegation** allows one principal to pass a sub−set of it's rights to another. A common example of limited delegation is to pass all rights from one principal to another, but to deny the ability to delegate those rights again. The delegated rights may also be limited in time. A **super−scheduler** is a special type of scheduler that can co−reserve and co−allocate multiple resources, and broker among multiple resources. A **trusted component** is an entity which is trusted to not abuse its given authorization.

2.2. Scenario 1

A user wishes to use a collection of Grid resources located in various administrative domains. For example, a user may have a large data set managed by a cache service like DPSS. She would like to use this data as input to a computational job, and then transfer the results to a tertiary storage system. Since in this scenario co−allocation of several resources is required, the user will ask a super−scheduler to schedule the job. The super−scheduler must schedule the appropriate cache machines, the necessary network resources to move the data from the cache to the compute server and from the compute server to the tertiary storage system, the compute server(s), and the tertiary storage system. Some of these resources may have their own local schedulers with which the super−scheduler will have to communicate. In particular, the network may be scheduled through the use of a bandwidth−broker that allows a network path to be treated as a single resource, comparable to a VLL (Virtual Leased Line). The compute resource may also be scheduled through a local scheduler.

Required security operations:

- The user must authenticate herself to the super−scheduler or it's security gateway. The super−scheduler or gateway may authenticate itself to the user. If the super−scheduler is a "trusted" component, it will have to authenticate itself to the user.
- The user must delegate some sub−set of her rights to the super−scheduler. In the absence of limited delegation the super−scheduler must be trusted to not abuse the delegated rights, i.e. the super−scheduler must be a trusted component. If a security gateway was used to authenticate the user, the super−scheduler must have a standard way of acquiring the the users delegated credentials from the gateway. Additionally, the user should have full control over the delegation process. To implement this, the user should be able to specify the delegated sub−set of rights, and to add (signed) policy information to this process, i.e. information used to approve whether the remote principal is

authorized to use the delegated credentials.

- The super−scheduler will determine the "best" computer, as defined by a predefined metric such as cheapest, or fastest on which to run the users job. This may involve the super−scheduler contacting the Information Service to identify the candidate hosts. It must then discover which of the candidate hosts the user is authorized to use, by either querying the Information Service or by contacting each potential host directly. Due to the confidentiality of account and allocation records at most sites, the super−scheduler must perform this operation using the delegated identity of the user.
- A similar operation is performed by the super−scheduler to determine the "best" cache machine(s).
- If the super−scheduler is treating the "network" as a single resource it will have to communicate with a bandwidth−broker to schedule the network resources. The super−scheduler will query the bandwidth−broker to determine if sufficient network resources exist between the compute server and the cache machine(s), and between the compute server and the tertiary storage system. The bandwidth−broker may have to query each of the network links on behalf of the user to answer this question. Consequently, a super−scheduler must have the right to delegate at least once. Considering the fact that network resources reserved through a bandwidth−broker will typically be allocated after the socket connection has been brought up by the user‘s job, the super−scheduler should have the ability to delegate the right of delegation too (see below). If sufficient network resources do not exist, the super−scheduler will pick new "best" compute servers and cache machines.
- The super−scheduler will now contact each of the resources in turn to begin processing the job. This again involves the super−scheduler using the delegated credential(s) to authenticate itself on behalf of the user, to the resource or local scheduler. It may also be necessary for the super−scheduler to delegate the user’s rights to some or all of the requested resources. The local scheduler for the compute server may need the user’s delegated credentials so that it may further delegate them to the compute server. The compute server could then use these credentials to authenticate on the users behalf to the tertiary storage system. The super−scheduler will delegate the users rights to the bandwidth−broker so it may schedule the individual network links along the path.
- The local scheduler will authorize the requested operation based on the users identity contained in the delegated credentials, and possibly further delegate the users credentials as discussed above.

2.3. Scenario 2

A user is running a remote experiment on a large scientific instrument such as an accelerator or microscope and wants to transfer the data back to his home site for real−time visualization. Large scientific instruments are most typically scheduled in advance. Thus is will be necessary to reserve the required network bandwidth and the visualization engine in advance to guarantee their availability at the time of the experiment.

Required security operations:

- As above, the user authenticates to the super−scheduler and delegates his rights to it. This allows the super−scheduler to make the reservations on his behalf.
- The super−scheduler then attempts to contact the requested resources and make the reservations. The super−scheduler will use the delegated credentials to authenticate itself to the resources, so that it may create reservations on the user's behalf. While attempting to reserve the network, the super−scheduler may have to delegate the user's rights to the bandwidth broker so it may reserve the individual network links.
- The local schedulers will authorize or deny the request based on the identity of the delegated credentials.
- The local schedulers will have to generate some type of "reservation handle" or "claim ticket". This handle must have two important properties; first it must be non−forgeable, and second it must be non−repudiable. Non−forgeable means the reservation handle for the resource could only be generated by that resource or its scheduler. Non−repudiation means that the resource or local scheduler should not be able to deny that it granted a reservation.
- At claim time the user will interact directly with the local schedulers or resource gateways to instantiate his reservation. The user must be able to authenticate to the resource and prove that it is the principal that made the reservation, or that the reservation was transferred legitimately.

2.4. Scenario 3

Consider a complex Grid application which consists of several interdependent computational steps. Some steps might be executed in parallel, while others depend on the results of the former execution and have to be processed after the required input data has been computed.

Now consider the fact that the user wants to submit this job mixture in one single step, either to a super−scheduler or an agent, taking care of the correct order of each computational step. The task of this entity is to wait for events, i.e. job completion, whereafter further subjobs are submitted on behalf of the user.

Required security operations:

- As above, the user authenticates to the super−scheduler (or agent) and delegates her rights to it. This allows the super−scheduler to make the reservations on behalf of the user.
- Following the user supplied job schedule, the super−scheduler attempts to contact the requested resources. The super−scheduler will use the delegated credentials to authenticate itself to the resource, it will allocate the subjob, and it will wait for completion.
- The local schedulers will authorize or deny the request based on the identity of the delegated credentials.

- As this job mixture might have a significant complexity, it might be hard for the user to specify a reasonable lifetime for the delegated attributes in advance. To reduce the risk of using long−term credentials the super−scheduler should have the right to renew credentials within a given time interval.

3. Delegation Requirements

As the above described usage scenarios demonstrate, the ability to transfer rights to another principal is a crucial building block for a Grid scheduling environment. The super−scheduler must be able to allocate resources on behalf of the user. Impersonation is one possible solution for this. However, a standard impersonation credential does not allow the server to whom it was presented to access the complete delegation chain, and is therefore vulnerable to misuse. Current practice to address this issue is to limit the impersonated principal in its right to delegate again.

While this solution is appropriate for direct allocation requests, initiated by the user, a super−scheduler delegation model should provide at least the following features:

- Local resources, or their security gateways, should be able to identify all of the principals in the delegation chain, and verify that each step of the delegation process was properly authorized. This implies the use of true delegation instead of impersonation. This information can then be used by the resource to make policy decisions about what delegated credentials to trust. For example, NASA Ames may be unwilling to accept a delegated credential that was formerly passed to a non−NASA site.
- The user should control what rights she delegates, and to whom they are delegated. This will be very difficult to achieve in a Grid environment where the user may not know where her job will run, nor exactly what rights it will need to complete. This leads to a situation where either the user grants more rights then necessary, which may lead to abuse, or grants too few rights which prevents the task from completing. It may be useful to consider abstracting specific rights into more general standard categories, e.g., "execute my job", "read file", "write file", etc. In general, the process of delegation should support the transfer of user policies, to control the use of the delegated rights.
- The user should be able to control how long her delegated credentials are valid.
- A scheduler or a super−scheduler should have the right to renew the credentials within a given maximum life time. This is to support long queuing times and job chains.

4. Author’s addresses

Keith R. Jackson

Distributed System Department
Lawrence Berkeley National Laboratory
1 Cyclotron Road MS: 50B−2239
Berkeley, CA 94702
Tel: (510) 486−4401
Fax: (510) 486−6363

Volker Sander
Central Institute for Applied Mathematics
Forschungszentrum Jülich GmbH
52425 Jülich
Germany
Tel: +49 (2461) 61−6586
Fax: +49 (2461) 61−6656