Security in Grid Computing

Tak-Lon Wu

B534 project 3

Computer Science Dept.
Indiana University
Bloomington, IN 47405
taklwu@indiana.edu

Abstract

Over last 10 years, security one of the important issue within the field computer science. Meanwhile, Grid computing has been rapidly growing with its powerful computational feature. As a result, researcher starts to deploy security within Grid computing in order to provide a better, safe, and trusted environment. Different architectures were proposed to dominate this field of studies such as Legion, Globus and CRISIS for WebOS. This paper will present a review and comparison of these security architectures.

1. Introduction

Security is an essentials topic in Grid Computing. A major task of this achievement is to provide a safe and trusted environment among this large-scale computational system. In the past 10 years, there is various architecture tend to accomplish this task, they are Legion [1], Globus [2], and CRISIS [3]. These different architectures have similar goals such as least privilege and access control, but both of them try to use a different methodology to achieve the same goals.

In this paper, we first make a short review of why a Grid environment needs security, and what kind of security is generally required. Then, we introduce these three systems and describe the main features of them. Furthermore, this paper indicates the
differences of them and base on these differences, we give a brief discussion. And finally, a short conclusion is given.

2. Why do we need security in Grid Computing?

In Large Scale Grid computing environment, requests are frequently sent through the entire system. Such requests properly are resources requirement, job assignment, process communication, and other interaction between different Grid machines. Before acting these actions, it is better to provide specific mechanism to protect the resource supported by each machines. As a result, security is an essential work. However, the security here does not only represent the traditional cryptography, they should include

- Integrity
- Privacy
- Least privilege & Access control
- Isolation
- Accountability

First, Integrity means resource provided by a Grid machine cannot be modified. Second, privacy represents the communication within the Grid environment should be safe (e.g. message encryption). Third, Least privilege & Access control is an important issue that a resource or process should be only utilize by a permitted object (e.g. process, user). Fourth, Isolation means the entire system must reminds safe even though one of the grid server is compromised. Least but not least, Accountability is a way to audit the entire system in order to modify the security need. The following section mainly focuses on how a system achieves Least privilege and Access control.

3. Three systems

In this section, the main features of three different systems are going to introduce, they are Legion [1], Globus [2], and CRISIS for WebOS [3]. The security goals of these systems are very similar, but the architecture is somewhere different to each other. Especially, Legion, which only provides a high level structure to developer, designs their system with different security concerns.

3.1 Legion

Legion is a Grid computing platform which combines a large scale of independently administered machine. The resources, include processors, database, user objects and other objects, are all distributed over a wide area network; each machine has its own administrative domain(s). The way that Legion combines these numerous components together by using a single, object-based metacomputer which achieve its primary goals, flexibility and site autonomy. One of the designs based on this single object-based is Security.

There is a Legion Runtime Library (LRTL) interface, which defines basic interfaces to services such as object creation, location, and deletion, to support the object-based
implementation. Furthermore, it defines two Core Object, Host and Vault. Host object is a process manager (e.g. ensure a user does not use more time than it allowed). Meanwhile, A Vault Object is an object that controls availability of stable storage (e.g. ACL of a file system). These two objects are encapsulated within a “Class Manager Object” which is responsible for managing the placement, activation, and deactivation of a given object(s). The security design also follows these two principles. A simple architecture is shown as Figure 1.

![Figure 1 Legion simple architecture](image)

As the goals of Legion are flexibility and site autonomy, it just provide few default implementation for security extension. First is a Legion Object identifier (LOID), which can identify an object within a Legion environment. The default security mechanism is using public/private (RSA) and certificate (X.509) key. Each service (resource) and user should hold a LOID to handle the communication. A user LOID should register to the system, and then enter the appropriate system group and ACL given by the resource provider. Meanwhile, a service (resource) will create a “credential to object” in order to perform action on a given user’s behalf. Second, ACLs is hold by each object itself. This mechanism is certainly distributed and can achieve the Isolation issue as it does not have centralized ACLs. Third, Legion is a software that runs on the existing OS with its own policies. It just needs to ensure that Legion mechanism cannot achieve a high level security issue than it is restricted. As a result, Legion provides a Process Control Daemon (PCD) to handle this problem. For those users who have an account on the local OS, PCD will run on his behalf. For other users (foreign request), PCD will create a generic account to create a minimal permission account in order to support the allowed action(s). This PCD is small and easily to monitor by local administrator and also match the “Least Privilege” requirement.

### 3.2 Globus

As Globus assumes that a Grid system is large and dynamic. The system should require all of the standard security function which includes integrity, privacy, authenti-
cation, access control and non-repudiation. However, as the paper [2] was written in 1998 and it was the first version of security issues of Globus, it focused on authentication and access control.

Globus is also a system that includes numerous trust domains. User, resource, process is a participant called subject (both global and local with a different name), whereas a resource that is being protected by security policy is called object. The process that a subject proves its identity to another subject with its credential is called authentication. After authentication, user can use/access an object on the local machine by authorization. In addition, all the access control decision of an object is made locally by the administrator.

Globus architecture1

As Globus need to achieve the authentication and authorization, it introduces a special mechanism, proxy. Figure 2 show a general architecture of Globus. As can been seen, there are two kinds of proxy in Figure 2, one is the user proxy and the other is resource proxy. A User Proxy (UP) is mainly responsible for user log-in and passing authentication message. It is a session manager that acts on behalf of a user for a limited period time. A Resource Proxy (RP) is an agent used to transfer the security operations between different administrative domains and handles local intra-domain mechanisms. In other word, RP deals with user authentication and local resource authorization and local resource/process allocation. Here, it should be mentioned that RP can pass the credential received from UP through other RP. In addition, as all the access control decision must follow the local ACIs, a global to local mapping is provided. It follows the same idea as the authorization, but calling a different method. And the global mapping name is store at each machine as a database. After such mapping, a user

1. Protocol 1, 2, 3, 4 shown in Figure 2 refers to the original protocol which mentions in Ian Foster et al [2]’s paper.
proxy could allocate resource on behalf as a local user. However, having such database store in each machine might be an information disclosure when a node is compromised.

With these mechanisms, a single sign-on function is achieved. This function is a significant different to others, although this might be a vulnerability that attacker can comprised one of the RPs to get the user authentication. But there is one more requirement, each actions performed by a RP should request to its user proxy to honor the request can be done or not. This step could mitigate the vulnerability.

3.3 CRISIS

The purpose of CRISIS [3] mainly acts as a component within WebOS. WebOS is a wide area application which aims to support network application and provides a basic function as a common OS services. And CRISIS is a part of design which handles the security issues including Redundancy, Least Privilege, Accountability, and Local Autonomy (Access control). Although it covers a lots of security issues, we focus on the authentication, authorization and least privilege.

CRISIS is an event-based security services that simply like Globus [2]. There is some Process Managers (PM) and a Security Manager, the former listens to a certain request, e.g. login and resource request; meanwhile the latter stores the credentials of a user/resource and validates the certificate transmitted by others. These PMs identify the administrative domain within a request, and then ask the Security Manager whether they have the privilege to access remote resource. Figure 3 shows an example of how a user accesses a remote file from a remote node. As can been seen, the process with a security domain will request a system call “open” to open a remote file. Then this

![Figure 3 CRISIS architecture with accessing a remote file](image-url)
process will send a message to Kernel which has a Vnode Layer to translate the system call to an appropriate service (here is WebFS server). The service will then acquire the credential information (public key with time limitation) from a local Security manager, and send a request to the remote resource node with this information. Once the remote service receives the request, its local Security manager will check those credential information with the Locally trusted on-line Agent (OLA) to validates the time limitation of this public key, or check whether is a stolen key. This OLA is trusted by local nodes and assigned by a root Certification Authority (CA), which signs all the identity certificates with a long time-out. In addition, this OLA takes the responsibility for signing short session certificates. If all the process is succeed, it return the value to the user process.

The authentication and authorization method is likely the same as Globus, a single sign-on mechanism. Meanwhile, it includes a User-level and Kernel-level mechanism match the least privilege rule. In addition, the CA and OLA achieve a hierarchical trust for the public key mechanism, which also make the system have redundancy (attacker needs to attack both the normal node and the CA/OLA).

4. Differences

<table>
<thead>
<tr>
<th></th>
<th>Legion</th>
<th>Globus</th>
<th>CRISIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Flexible</td>
<td>Extendable</td>
<td>Fixed</td>
</tr>
<tr>
<td>Authentication</td>
<td>LOID</td>
<td>User and Resource Proxy, single sign-on mechanism</td>
<td>Service Process and Security manager, single sign-on mechanism</td>
</tr>
<tr>
<td>Authorization</td>
<td>LOID</td>
<td>User and Resource Proxy</td>
<td>Service Process and Security manager</td>
</tr>
<tr>
<td>ACL</td>
<td>Object-base ACL</td>
<td>Using local machine ACL which checks with local Resource Proxy</td>
<td>Using local machine ACL which check with local Security manager</td>
</tr>
<tr>
<td>Authentication session</td>
<td>Maintain by user/developer</td>
<td>Long term session by Resource Proxy or short term by User Proxy</td>
<td>Public key with session timeout</td>
</tr>
<tr>
<td>Isolation</td>
<td>Host and Vault Objects use their own dedicated local accounts to ensure isolation from other user objects.</td>
<td>N/A</td>
<td>CA/OLA redundancy</td>
</tr>
<tr>
<td>Least privilege</td>
<td>Using PCD to act as different level of account privilege</td>
<td>N/A</td>
<td>Using user-level and kernel level system call to reduce the privilege of a user/remote process</td>
</tr>
<tr>
<td>Accountability</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1 the difference between Legion, Globus and CRISIS

The differences between these three architectures are shown in Table1. As mentioned in previous section, these three systems are really similar to others. However, as

2. WebFS is a global file system similar to NFS.
shown in Table 1, we could see that Globus is the one that serve with the fewest security features. The reason might due to this is a first version of this security architecture, and this version focuses on the single sign-on mechanism. In addition, CRISIS likely provides a completed implementation for all security issue. Meanwhile, Legion seems to support all the security features, however, the detail implementation of Legion must be done by developer. Finally, as we only focus on the Authentication, Authorization, ACL, Least privilege, and Isolation, some other different might not be mentioned with our study.

5. Discussion

After comparing the differences, and skip ahead to nowadays, we might know that why these architecture are still using or disappeared. First of all, single sign-on is a good mechanism to reduce the user overhead to login to different service provide by a Grid System, although this mechanism will raise the complexity of a system. Second, keeping ACLs control to a local administrative domain helps a Grid system being more flexible and is more suitable for realistic requirement. Third, having time session for an authentication/authorization action could enhance the security for being attack by a compromised node. Meanwhile, this feature sometimes helps a Grid system achieve isolation. Fourth, the CA/OLA hierarchal trust mechanism is a high-level skill that avoid attacker taking control of a Grid node. However, this might increase network traffic as each node must check the CA/OLA before running a task. Last but not least, the least privilege is really important issue in today’s security, even though Legion and CRISIS seems to achieve this goal. For instance, PCD runs as root permission, once if attacker handle this PCD process, the system will break. Therefore we should add one more certificate mechanism before this PCD gain the root permission such as “sudo” command in Ubuntu Linux.

Finally, as I know, Globus tookit is the only one still widely used by researcher and enterprise. And reason is probably as its simple single sign-on mechanism. Legion’s people formed a company call Avaki Corporation in 1999, but it was abandoned in 2001 by the main research group. After that, no one keeps updating this architecture. CRISIS does not have much information online, but as the paper told us it was test between University of California, Berkeley, University of Washington, and University of Texas. And the reason that CRISIS did not widely use by enterprise might due to the original motivation of its project is a component of WebOS.

6. Conclusion

This paper briefly give a introduction among three security architecture Legion [1], Globus [2], and CRISIS [3]. Their common features include authentication, authorization, least privilege, isolation, and Access control. Focusing on these features, we shortly indicate the differences between these systems. Furthermore, we discuss the pro and cons among these systems, and make a conclusion why some of these systems are now widely used or seldom taken as research tools.
Finally, in my opinion, some of these features of an individual architecture could be combined together. For instance, the CA/OLA hierarchal trust mechanism could be added to a Globus based system in order to achieve Isolation. In addition, the flexibility of Legion could provide researcher a test environment of new security issues. Having such review of security in Grid Computing helps me understanding in different major studies, security concerns depend on vary requirement.

Reference

