Security in Grid Computing

Muhammad Asif Habib* and Michael Thomas Krieger**

Johannes Kepler University, A-4040 Linz, Austria

Abstract. Grid computing provides high computing power, enormous data storage, and collaboration possibilities to its users. In the networked access to computation with a single-sign-on system as the portal to the possibilities of world wide computing grids security plays an important role. This paper provides an overview about the state-of-the-art of security in current grid technologies and a discussion about possible weaknesses which have to be considered at the current approach.

1 Introduction

When in 1969 the project ARPANET, which subsequently became the internet, went online Leonard Kleinrock was cited in a press release from UCLA that *computer networks are still in their infancy. But as they grow up and become more sophisticated, we will probably see the spread of computer utilities which, like present electric and telephone utilities, will service individual homes and offices across the country* [1]. 20 years later with the introduction and standardization of HTTP and HTML [2] the internet and the world wide web started to spread out to every home.

The growing number of devices and thus mostly unused resources connected to the internet triggered many different ideas to share available computing and storage resources. In 1998 Ian Foster and Carl Kesselman defined in the book *The Grid: Blueprint for a New Computing Infrastructure* a computational grid as *a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.* [3] This definition was subsequently refined.

In general, grid computing provides users with the ability to divide and spread large computations across multiple machines as well as access to distributed storage and collaboration possibilities within virtual organizations [4]. As in [5] stated, grids allow *the simultaneous use of large numbers of resources, dynamic requirements, use of resources from multiple administrative domains, complex communication structures, and stringent performance requirements.*

The coordination, administration, and subsequently also the billing of resource usage in a grid is not only challenged by the decentralized control and ownership of the computing and storage structure but also by the aim to provide an efficient and usable environment to its users. Current grid middleware like

* habib@fim.uni-linz.ac.at

** mk@druide.at

the Globus Toolkit[1], legion [2], gLite[3], or BOINC[4] offer a basic software infrastructure and tools for hiding the complexity of the heterogeneous infrastructure of a grid from the users, for easing the administration and configuration of the participating systems, for coordinating and monitoring available resources, and for providing a basic layer for developers to create grid applications.

In the following a closer look will be taken at security threats, countermeasures, and detection in grids and how different grid middleware solutions approach these issues.

2 Components of Grid Middlewares

The decentralized approach of grids, the flexibility and the heterogeneous nature of their infrastructure, and the aim of being a general purpose system are a challenge to grid middlewares to provide a manageable distributed, secure, stable, and high quality-of-service system to its users. For achieving this most approaches of grid middleware are split up into the following components [6]:

- *basic middleware*: This module provides the basic abstraction layer from the system integrated in the grid and also the API for developing and running applications on the grid.
- *authentication and authorization system*: This module is responsible for the authentication and authorization of users, virtual organizations, and processes accessing the grid.
- *workload management / scheduling*: This module manages the scheduling, distribution, and prioritization of jobs and processes running on the grid.
- *data management*: This module manages the data storage and also the access to data on the grid.
- *fabric management*: This module provides tools for installation management of grid applications and basic resource management, monitoring, and configuration.
- *information system*: This module collects available information about the grid like availability and status of resources, the job queue and the status of active jobs, information about users and virtual organizations, etc. These systems allow to monitor the grid and also provide tools allowing users to interact with their submitted jobs.

Due to the characteristic of grids each of these components provides a challenge for security in informations systems to prevent manipulation, misuse, unauthorized access, denial of service, hijacking, stalling of processes, and stealing of information stored on the grid, computing power provided by the grid, and in general devices connected to the grid. The large and dynamic user population and resource pool, the dynamic acquisition and release of distributed resources

[1] http://www.globus.org/
[2] http://legion.virginia.edu/
[3] http://www.glite.org/
[4] http://boinc.berkeley.edu/

during computation, and the different authentication and authorization mechanisms in the heterogeneous environment of grids require a broad view on security and contains many security challenges [5].

3 Security areas

Each category of grid middleware modules is confronted with security issues; either with issues concerning only its category or issues concerning several categories. But as soon as one module is compromised or provides a security hole the whole grid infrastructure may be compromised by attackers, which have the chance to hijack the grid and/or get into a distributed computing infrastructure which is designed for high performance computing. Starting out with taking care of user authentication in grids [7][5] the security working group of the *grid forum* started to list issues, which also have to be addressed when looking at the security of grids [8].

3.1 Authentication and authorization

Most security considerations in grids are focused on the authentication and authorization to access the available resources on the grid. From a usability point of view to access the grid and all its resources users should enter their login credentials only once. This single-sign-on approach to access the heterogeneous and distributed environment of grids is a corner stone of the success and further growth and distribution of grids [7][5]. Using public key infrastructure (PKI) based on X.509 certificates has become the standard for grid middleware – like Globus or gLite – to implement the single-sign-on approach. The usage of this PKI establishes a mutual trust relationship between the user and the entry point to the grid allowing not only the grid to check the user's certificate but also vice versa allow the user to verify the entry to the grid via it's certificate. [6][9]

In addition to this basic authentication the middlewares Globus and gLite use a user proxy approach to delegate the credentials to the systems either used for computations by the user or the user's processes or containing data required by the user or the user's processes. Using a proxy the user delegates the rights to it which again can delegate the rights to processes started by the user and needing to access other systems of the grid infrastructure. To avoid the exposure and publication of the user's credentials the proxy uses its own credentials which are only valid for short period of time, usually for about 12 hours. [10]

On the systems themselves the grid users are mapped to local user accounts, which allow the execution of the requested jobs and access to data necessary for the execution of the request. In Globus and gLite this user-mapping is based on *gridmap-files*. [11][7]

In Globus and gLite the access to resources can also be limited by requiring users to be members of virtual organizations (VO). By restricting the access to systems of the grid infrastructure to special VOs, only members of those VOs are authorized to access them. [12]

3.2 Scheduling

The scheduling of jobs and managing a job's access to data, especially if the executed process is very data intensive [13], is an important topic in grid computing. Processes running in grid environments do not only require CPU time but also bandwidth and data storage, which should be reserved for the processes. Due to the structure of grids and resources which are managed, distributed scheduling of tasks improves the scheduling performance and according to [14] *makes a system portable, secure, and capable of distributing scheduling workload among an array of computational sites in the system.*

In [15] a scheduling approach is presented, where the discrepancy between the requested security levels and offered security levels influences the scheduling of jobs on the grid. The mapping from global user accounts to local user accounts on computing elements of a grid may lead to such deficiencies, which have to be considered at scheduling time, so that sensitive data or computations may not be modified or accessed by unauthorized users.

When looking at different grid middleware implementations, not all of them support security- and policy-based scheduling. Although Globus and gLite use GSI as a basic layer for all processes and users have the ability to specify several conditions which have to be met by resources to decide which ones should be used [6] the scheduling is by default based on the information specified by the users in the job description file and the user's virtual organization (VO). BOINC, as a high throughput grid systems, lets the owner of the resource decide at which project he wants to participate and all available resources are used. In contrast to Globus and gLite where the users are the ones accessing others' resources and thus have to trust the resources and the infrastructure, when participating at BOINC the users have to trust the software and data installed and transfered to them. Regarding security, the scheduler server of BOINC[5] checks, whether a resource — a user's computer — is reliable or not and sends the job request.

3.3 Execution

After a job is submitted and scheduled for execution it is submitted to the designated resource/computing element for execution. Several security aspects have to be considered at this stage. From an administrator's perspective the job should have no possibilities to do any harm to the resource it is running on. It should not be able access data and other jobs it is not allowed to access as well as it should not be able to consume so many resources on the computing element that other locally originated jobs starve due to resource shortage. Several methods like application-level sandboxing, virtualization, user-space sandboxing, or flexible kernels can be used to protect data on the computing element.[16]

Grid middleware like Globus or gLite only allow accountable users to submit jobs, develop new applications, and upload and access those jobs and applications. On computing elements gLite uses account-based sandboxing using

[5] http://boinc.berkeley.edu/trac/wiki/SchedMatch

Workspace management service (WMS) from Globus. In general WMS allows several levels of sandboxing, from account-based sandboxing by creating an user environment on the fly to full virtualization by booting a user specific virtual machine.[17]

Condor for example uses different environments (standard, vanilla, grid universe, etc.). Although as described in [18] sandboxing based on virtualization techniques may provide a more secure approach to keep insecure applications from harming the computing element.[19]

In contrast to these grid middlewares BOINC[6] does not use PKI for user authentication on the BOINC grid, which allows easy integration of resources. To prevent malicious executable distribution BOINC uses code signing, which allows only signed code to be executed on the computing elements. Each executable is checked before being executed on the computing elements, which also allows malicious code to be traced back to its originator. In addition to this account-based sandboxing is used on the computing elements, which prevents the grid applications to access other data stored on the computing elements. Account-based sandboxing only works if the privileges are set correctly. If the rights of the account created by BOINC are modified by the owners of the computing element for sure private information about the owner could be acquired as the resources used by BOINC applications are mostly privately owned computers.

3.4 Data Access and Management

Besides distributing and parallelizing computation on the grid the access to and storage of data in the heterogeneous environment of a grid also provides many challenges regarding distribution, replication, and performance as well as concerning security issues. Globus and gLite use GridFTP as transportation protocol, which is a FTP solution, is based on GSI, and uses transport layer security (TLS) for securing the file transfer between clients, storage elements, and computing elements.[11][6]

Besides GridFTP Globus also provides secure file transfer via webservices: reliable file transfer (RFT) service. In contrast to GridFTP, where the file transfer's state is lost in case of a failure of the client, RFT persists the state of the file transfer in reliable storage.[20]

gLite also offers the possibility of file I/O from computing elements to storage elements based on the Remote File Input/Output protocol (RFIO) — with an insecure and a secure (gsirfio) version available — and GSI dCache Access Protocol (gsidcap) without a complete file transfer. But in general when running a process a user specifies an input sandbox where the data is read from and an output sandbox where the output of the process is placed. Both sandboxes typically have to be owned by the grid user starting the process. All references to files on the grid which are read and/or written by grid applications are stored in a file catalogue — the LCG File Catalogue (LFC) — which also contains the access control list (ACL) for each file.[6]

[6] http://boinc.berkeley.edu/trac/wiki/SecurityIssues

Besides plain text storage on storage elements also encrypted storage is possible in gLite as described in [17]. The solution stores the encryption key distributed on several key servers and whenever a user or a process started by the user requests encrypted data, this key is retrieved and only on the computing element the data is decrypted for processing. Thus at two points of the grid architecture the data is available in plaintext: on the client side and on the computing element, where it is used as plaintext in memory and possibly in a temporal file on the disk.[17]

Apart from pure file based data access, storage, and management also a generic data access interface for grids is available. The project Open Grid Services Architecture Data Access and Integration (OGSA-DAI) is aimed at providing a web service based access to any possibility of accessing and storing data regardless of how the data is stored, whether the data is stored in databases, in structured, semi-structured, or unstructured file formats. OGSA-DAI is built upon the grid security infrastructure and requires web service based authentication (see [11]) to access data from data sources available via OGSA-DAI.[21]

3.5 Fabric Management

As grids are not part of one but are distributed over several administrative domains, installed services, configuration management, and update strategies vary widely. Therefore, the grid middleware has to compensate on one hand the distribution strategy of jobs to resources where the required services are available and on the other hand the backwards compatibility to older still running versions. Depending on the administrator of an administrative domain software updates are installed sooner or later, configurations regarding the security of the local system and/or the whole grid infrastructure have been applied correctly or not, etc.

Those mostly human factors currently are not compensated by current grid middlewares. For gLite EGEE[7] has compiled security best practices and guidelines (see [22]) which include tips for update, configuration, and system monitoring. Apart from these advisories gLite also includes the software installation, update, and configuration management tools quattor[8] and YAIM[9], which allow administrators to manage several nodes and apply configuration changes and updates to supported nodes at once.

Advisories and updates to the Globus Toolkit which should be installed are published on the website of the Globus Toolkit[10]. BOINC does not provide automatic update management either, although with the help of the BOINC manager it is possible to apply configuration modifications from a BOINC manager application to all client nodes which are configured to listen to this management application[11].

[7] http://www.eu-egee.org/
[8] http://quattor.web.cern.ch/quattor/
[9] http://yaim.info/
[10] http://www.globus.org/toolkit/advisories.html
[11] http://boinc.berkeley.edu/wiki/index.php/Controlling_BOINC_remotely

Overall centralized management tools for applying configurations to multiple grid nodes within one administrative domain are available, although the update process has to be triggered manually when using current grid middleware solutions.

3.6 Information and monitoring

Collecting information about available and used resources — computing and storage resources — , the status of jobs, active services, etc. is a vital part of managing and also using a grid infrastructure. Each grid middleware has tools to collect information provided optionally by computing and storage elements and information which must be available to schedulers and make this information via a public interface available to administrators and users of the grid. gLite for example includes the web information and monitoring tool GridICE[6], which can provide information about available memory, number of CPUs, storage size, etc. for computing and storage elements, which is useful information for administrators and users, but also possibly useful information for attackers.

At BOINC, if the owner of computing elements allows the publication of the information, it is possible to find out hardware and software information about the computing elements a user has the BOINC middleware installed on; for example the number and type of CPUs, size of memory, size of cache, and the active operating systems (OS version, updates).

Apart from this detailed information about resources of the grid infrastructure, which is publicly available on websites, grid portals also provide the possibility to submit jobs and interact with them. In addition to security issues of the grid middleware itself these grid portals introduce security issues of web applications to the grid infrastructure. Although *breaking into a Grid (...) may not necessarily allow the attacker access to backend Grid resources*, but as most grid portals allow users to access grid resources and manage their credentials, monitor and maybe even interact with their running jobs, breaking into grid portals provides the same rights to an attacker as the grid user has on the grid portal. Generally speaking, providing access to and control over grid resources to users via a web portal increases the security risk of a grid infrastructure.[23]

4 Conclusion

Due to the heterogeneous environment, the distributed infrastructure, the decentralized administration and monitoring, and the available computing power in grids security issues in grid computing are manifold. Apart from security issues on single grid elements, security issues of the grid middleware may affect all connected grid elements. Therefore, constant monitoring, vigilant distributed intrusion detection[24], thoughtful rights management, regular updates, etc. are even more necessary than in other computing environments.

Basic tools for monitoring and rights management are available. Still progress has to be made regarding data protection and data privacy in grids and

sandboxing of processes on computing elements. Many steps are necessary to provide a secure grid environment to its users and maybe even more to move the grid from mainly scientific applications to private and industrial application of publicly available grid computing and manage the step with grid computing, which the World Wide Web has achieved in the 1990ies.

References

1. Kleinrock, L.: An internet vision: the invisible global infrastructure (2003)
2. Berners-Lee, T., Cailliau, R.: Worldwideweb: Proposal for a hypertext project. Technical report, CERN (1990)
3. Foster, I.: What is the grid? - a three point checklist. GRIDtoday **1**(6) (July 2002)
4. Humphrey, M., Thompson, M.R.: Security implications of typical grid computing usage scenarios. In: HPDC '01: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing, Washington, DC, USA, IEEE Computer Society (2001) 95
5. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational grids. In: CCS '98: Proceedings of the 5th ACM conference on Computer and communications security, New York, NY, USA, ACM (1998) 83–92
6. Burke, S., Campana, S., Lorenzo, P.M., Nater, C., Santinelli, R., Sciaba, A.: gLite 3.1 user guide. Members of the EGEE-II Collaboration. 1.2 edn. (April 2008)
7. Butler, R., Welch, V., Engert, D., Foster, I., Tuecke, S., Volmer, J., Kesselman, C.: A national-scale authentication infrastructure. Computer **33**(12) (2000) 60–66
8. Humphrey, M., Thompson, M.: Security implications of typical grid computing usage scenarios (2001)
9. Park, N., Moon, K., Sohn, S.: Certificate validation service using xkms for computational grid. In: XMLSEC '03: Proceedings of the 2003 ACM workshop on XML security, New York, NY, USA, ACM (2003) 112–120
10. Novotny, J., Tuecke, S., Welch, V.: An online credential repository for the grid: Myproxy (2001)
11. Welch, V.: Globus toolkit version 4 grid security infrastructure: A standards perspective. Technical report, Globus Alliance (2005)
12. Kim, B.J., Hong, S.J., Kim, J.: Ticket-based fine-grained authorization service in the dynamic vo environment. In: SWS '04: Proceedings of the 2004 workshop on Secure web service, New York, NY, USA, ACM (2004) 29–36
13. Xue, Y., Wan, W., Li, Y., Guang, J., Bai, L., Wang, Y., Ai, J.: Quantitative retrieval of geophysical parameters using satellite data. IEEE Computer **41**(4) (2008) 33–40
14. Xie, T., Qin, X.: Enhancing security of real-time applications on grids through dynamic scheduling. In Feitelson, D.G., Frachtenberg, E., Rudolph, L., Schwiegelshohn, U., eds.: JSSPP. Volume 3834 of Lecture Notes in Computer Science., Springer (2005) 219–237
15. Xie, T., Qin, X.: Security-driven scheduling for data-intensive applications on grids. Cluster Computing **10**(2) (2007) 145–153
16. Chakrabarti, A., Damodaran, A., Sengupta, S.: Grid computing security: A taxonomy. IEEE Security and Privacy **6**(1) (2008) 44–51
17. EGEE JRA3 team: EGEE Global Security Architecture for web and legacy services, EU Deliverable DJRA3.3. (2005)

18. Santhanam, S., Elango, P., Arpaci-Dusseau, A., Livny, M.: Deploying virtual machines as sandboxes for the grid. In: WORLDS'05: Proceedings of the 2nd conference on Real, Large Distributed Systems, Berkeley, CA, USA, USENIX Association (2005) 2–2
19. Butt, A., Adabala, S., Kapadia, N., Figueiredo, R., Fortes, J.: Finegrain access control for securing shared resources in computational grids (2002)
20. Globus Alliance: GT 4.0 Reliable File Transfer (RFT) Service. (March 2008)
21. The University of Edinburgh: OGSA-DAI 3.0 Documentation. (2007)
22. Wartel, R., Forti, A., Rumler, R., Kouril, D., Bermejo, C.F.: Security best practice (2008)
23. Vecchio, D.D., Hazlewood, V., Humphrey, M.: Evaluating grid portal security. In: SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing, New York, NY, USA, ACM (2006) 114
24. Jarmolkowicz, M.W.: A grid-aware intrusion detection system. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby (2007) Supervised by Prof. Robin Sharp, IMM, DTU.