FERMILAB-PUB-07-483-CD

Addressing the Pilot security problem with gLExec

I Sfiligoi[1], O Koeroo[2], G Venekamp[2], D Yocum[1], D Groep[2] and D Petravick[1]

[1]Fermi National Laboratory, Batavia, IL 60510 , USA
[2]NIKHEF, 1098 SJ Amsterdam, The Netherlands

E-mail: sfiligoi@fnal.gov

Abstract. The Grid security mechanisms were designed under the assumption that users would submit their jobs directly to the Grid gatekeepers. Many groups are however starting to use pilot-based infrastructures, where users submit jobs to a centralized queue and are successively transferred to the Grid resources by the pilot infrastructure. While this approach greatly improves the user experience, it does introduce several security and policy issues, the more serious being the lack of system level protection between the users and the inability for Grid sites to apply fine grained authorization policies. One possible solution to the problem is provided by gLExec, a X.509 aware suexec derivative. By using gLExec, the pilot workflow becomes as secure as any traditional one.

1. Introduction

The Grid security mechanisms[1] were designed under the assumption that users would submit their jobs directly to the Grid gatekeepers. However, for the past decade direct submission has never been the main submission mechanism for the majority of the users. With the proliferation of Grid sites, most users prefer to submit their jobs to some sort of intermediate queue and have a workload management system (WMS) optimize the distribution of their jobs among the Grid sites.

Over the past few years, many groups have started to use pilot-based WMSes, prized for their ability to keep Grid-wide fair share between their users. These systems do not submit the jobs directly to the Grid gatekeeper, but send only so-called pilot jobs. Once a pilot job starts on a Grid resource, it will fetch a real user job and execute it. The traditional Grid authentication, authorization and mapping infrastructure is not used in the process, which has deep security implications, both for Grid sites and the WMS administrators.

gLExec, a X.509 aware suexec derivative is presented in this paper as a possible solution to the security problem. The advantages and the limits of this approach are described, as well as the experience gained from the initial deployment at the Fermi National Laboratory.

2. The security problems of the Pilot infrastructure in the current Grid model

In the current Grid model, the site Grid gatekeeper is responsible for authenticating and authorizing a user and to convert the user's Grid identity into a local one. If a user is accepted, his job is submitted to a local batch system using this local identity. When the job reaches a worker node (i.e. a compute resource) it runs using this local identity. Since different users are typically mapped to different local accounts, system level protection mechanisms prevent different user's jobs to interfere with each other. See figure 1 for an overview.

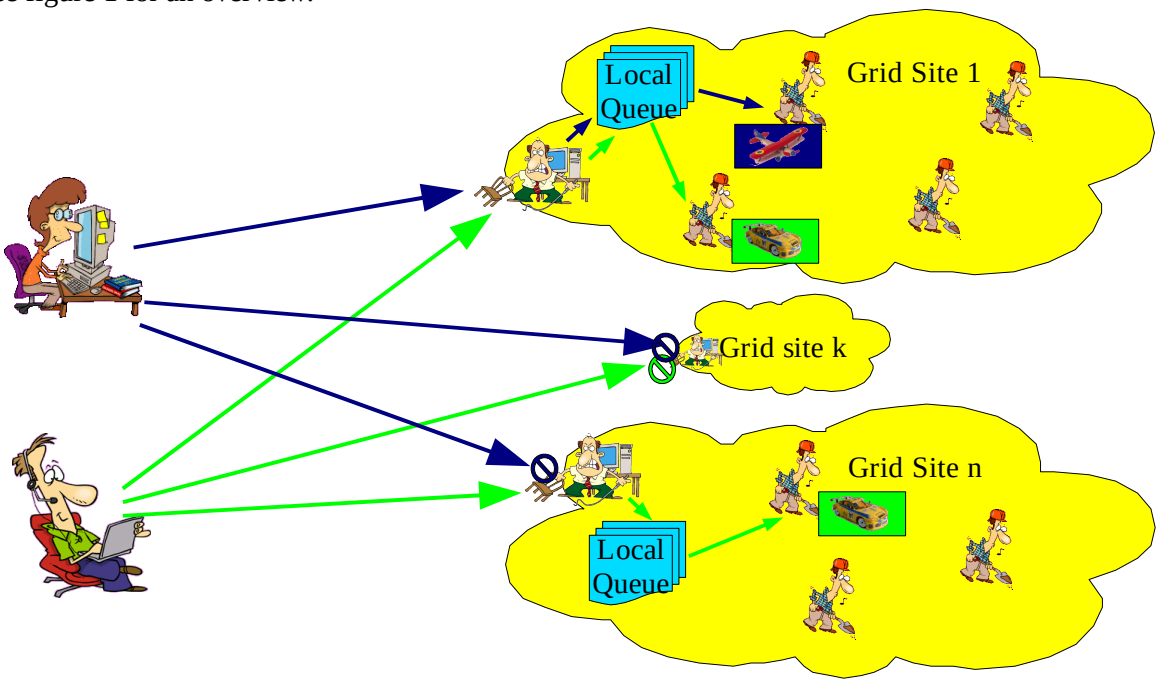

Figure 1. Traditional Grid security overview

When pilot based WMSes are used, the forkflow is quite different. Here, users submit jobs to a WMS specific batch system, and their presence triggers the pilot factory to send pilot jobs to various Grid sites. The pilot jobs then follow the path user jobs would, and eventually they start on a set of worker nodes. Each pilot job then pulls a user job and executes it. See figure 2 for an overview.

This modus operandi has several security drawbacks:

- From the user point of view, all the system level protections are lost, since all users run under the pilot local identity. Any other job of the same WMS can interfere with his own job, if running on the same worker node.
- From the WMS administrator point of view, there is no system level protection against malicious users, since both the pilot and the user run under the same, pilot local identity. A malicious user can impersonate the pilot infrastructure, completely breaking any security model the the pilot infrastructure has put in place.
- From the Grid site point of view, most authentication and authorization mechanisms are lost. The site can only allow allow all users to run, by allowing the pilot job, or prevent all of them, by denying entrance to the pilot jobs. Moreover, if pilot jobs are allowed, no user level accounting is available.

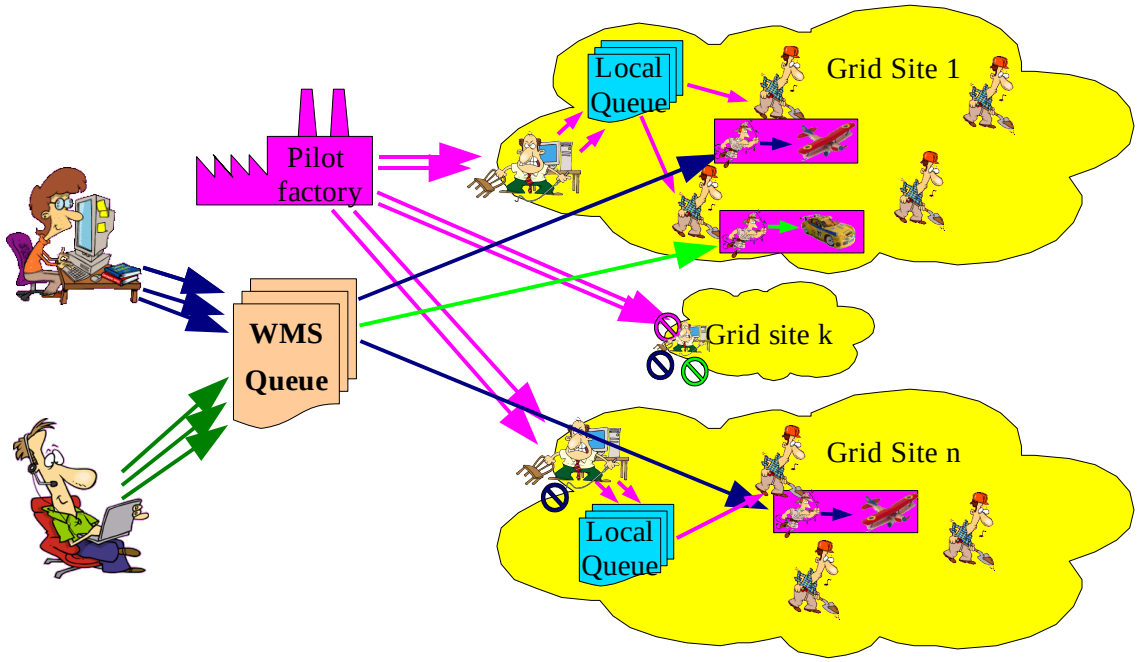

Figure 2. The Pilot workflow in the traditional Grid security environment

It is worth noting that the pilot based WMSes are not introducing the above mentioned security problems because they want to; they do it because there is no site provided security tool on the worker nodes. If there were one, most pilot based WMS infrastructures would be more than happy to use it.

3. Pilot security with gLExec

One security tool that can be installed on the worker nodes is gLExec, a X.509 aware derivative of the Apache suexec. gLExec is a lightweight privileged executable that, given a X.509 proxy certificate, authenticates and authorizes the user and runs the associated payload under the appropriate local identity.

Once gLExec is integrated in the the Grid security infrastructure and deployed on the Grid compute resources, pilots can use it to run users' jobs under the appropriate local identities, solves all the above mentioned security concerns:

- Users get back system level protection, shielding them from other users interference.
- The pilot jobs also gets back the system level protection, shielding the WMS security infrastructure from malicious users.
- The Grid site administrators set the local policies, and are guaranteed that unwanted users jobs will not run on their resources. Local accounting can now properly distinguish resources used by pilots from those used by the final users.

See figure 3 for a schematic overview. Please notice that gLExec behaves just like a gatekeeper, just running on each and every worker node.

4. Remaining problems with the Pilot infrastructure

While gLExec can solve the basic security issues introduced by the use of pilot infrastructure, there are still several concerns that remain unsolved.

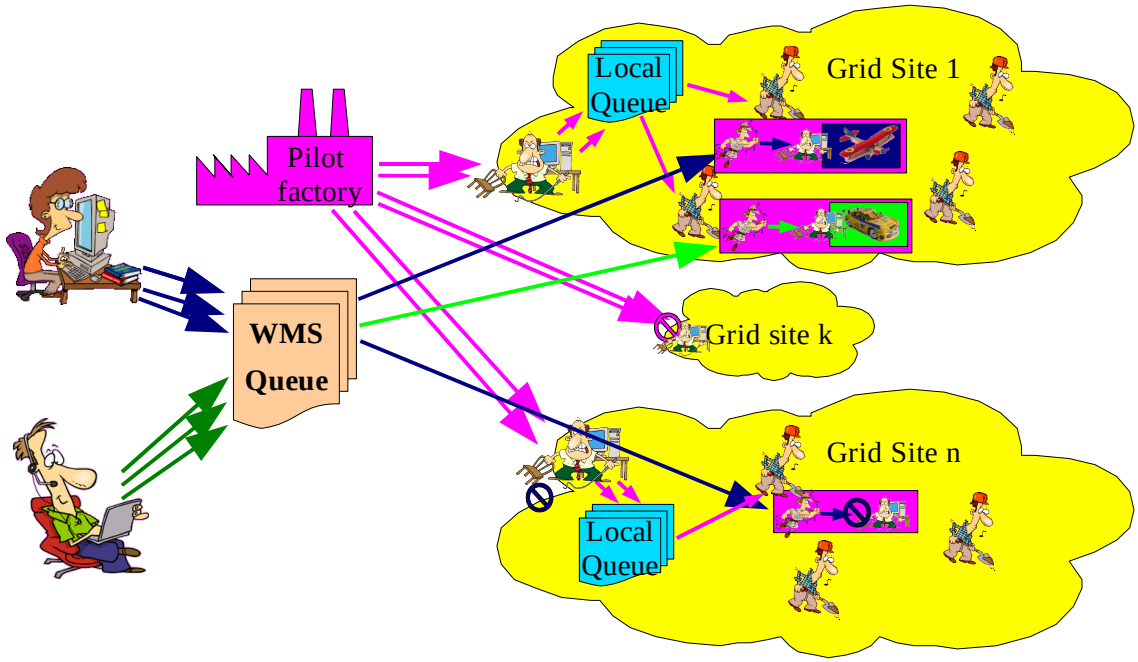

Figure 3. Pilot workflow with gLExec

4.1. Security of the Pilot infrastructure itself

Grid sites are rightfully worried about the trustfulness of the pilot infrastructure. The pilot WMS software was not provided, or even reviewed, by any Grid body the site trusts, and there is no explicit trust relationship in place between the WMS software providers and the Gird site, neither is there one between the WMS administrators and the site. gLExec by itself obviously cannot help solve this problem.

To be correct, the problem of trustfulness is not limited to pilot based WMSes. Any WMS that handles user jobs has the same issues. So the problem, although pertinent, is outside the scope of this paper.

4.2. Ensuring pilots do use gLExec

Installing gLExec does not force pilot jobs to use it. Moreover, it is technically impossible to force pilot jobs to use it, since a pilot job is technically no different than any regular user job. So Grid sites are rightfully worried about this.

However, the problem can be solved at the policy level. A site policy that holds the pilot job owners responsible for every action a user job performs unless they use gLExec can provide the Grid site with strong legal protection and be a strong incentive for pilot based WMS administrators to use gLExec compliant pilot jobs.

4.3. Nested gLExec invocations

Pilot jobs need to control their children and do proper cleanup after they are done. When gLExec is used, pilots need to invoke gLExec to perform these actions. However, if any child uses gLExec itself, the pilot may not be able to control those grandchildren.

The problem is that the pilot does not own the credentials that were used by its child to invoke gLExec, so it cannot use gLExec to act upon the processes and files resulting from such action. Although the pilot has access to all the files of the child, it would have a hard time find out which one was used to invoke gLExec. Moreover, a malicious users could delete such credentials after calling gLExec, making the process completely impossible. See figure 4 for an overview.

There is currently no knows solution for the problem of nested gLExec invocations. However, no known uses cases exist today, so nested gLExec invocations could be banned until a solution is found. To prevent abuse, a privileged monitoring tool could detect and kill any process trying to use nested gLExec invocations.

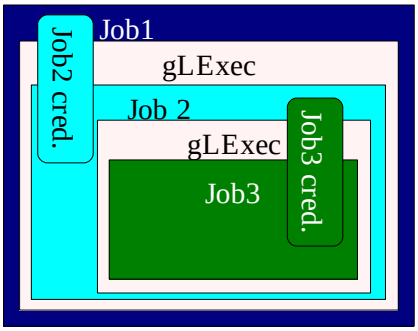

Figure 4. Nested gLExec invocations

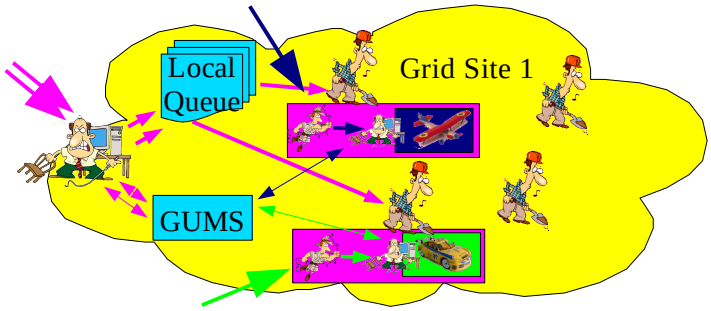

Figure 5. gLExec with GUMS

5. gLExec deployment

gLExec has been deployed on the Fermi National Laboratory (Fermilab) worker nodes since October 2006, and has been used by the CDF pilot based WMS, called the GlideCAF since then.

Fermilab is an OSG site, and thus uses GUMS[2] for user authorization and mapping. GUMS is a central service that can serve concurrently several clients. While its original goal was to unify the mapping between compute resources and data handling resources, having a centralized service proved to be extremely useful with the introduction of gLExec. Having to configure each and every worker node's policy would have been a nightmare, with GUMS, installing gLExec adds almost no additional complexity to the system. See figure 5 for a schematic view.

gLExec monitoring has also been interface to GRATIA, the OSG accounting system, providing local system administrator proper accounting of final user activity.

gLExec, together with the GUMS plugin been incorporated into VDT as of version 1.8. Moreover, OSG is planning on integrating it into its stack in the upcoming version 0.8.

Conclusions

Pilot jobs are becoming increasingly popular in the Grid world and the traditional Grid security mechanisms are not able to handle them. Installing gLExec on the worker nodes provides a way to solve most of the security problems introduced by the pilot concept.

gLExec has been integrated with the OSG security mechanism and has been extensively tested at Fermilab. The expected wide deployment on OSG will provide a first reasonably secure pilot friendly Grid environment.

References

[1] Foster I and Kesselman C 1998 *The Grid: Blueprint for a New Computing Infrastructure*. (San Francisco, CA: Morgan Kaufmann Publishers)

[2] Lorch M, Kafura D, Fisk I, Keahey K, Carcassi G,Freeman T, Peremutov T, Rana A S 2005 *Authorization and Account Management in the Open Science Grid* Grid Computing, 2005. The 6th IEEE/ACM International Workshop on 8