



GridSite security toolkit

1. GridSite 0.1 and GridPP

The GridPP Project began in 2000 as "The UK HEP Grid", with a collaboration of particle physicists and computer scientists at a dozen sites around the UK, and the intention to build a computing and data grid connecting resources and physicists. The researchers and system managers involved had worked together before as part of experimental particle physics collaborations, but new collaborative structures and tools were required to link staff at sites together as part of the grid infrastructure, rather than via experiment-based collaborations.

By 2001, the UK HEP Certification Authority had been set up and was issuing X.509 user certificates to all members of the project, which could be used for grid job submission. Since these certificates were also compatible with standard web browsers across Unix, Windows and Mac platforms, we were able to design a web-based management system using the same credentials. "GridSite" was implemented as a CGI filter, which was run by the Apache web server for each request for an HTML web page or graphics file.



Access control was by hidden files each directory, listing the X.509 DNs of users with read or write access. Users with write access could edit pages via web forms generated by the GridSite CGI executable.

2. GridSite 1.0 and GACL

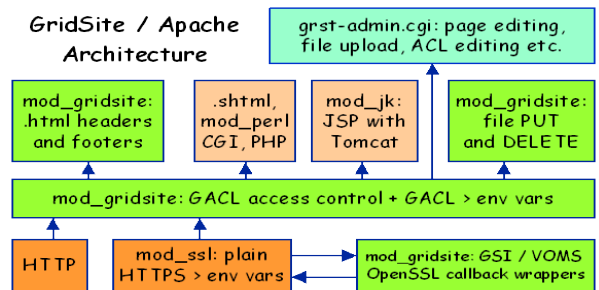
Although the CGI form of GridSite worked well, it imposed an overhead in that the CGI process had to be restarted for each page or image request, and it could not be used to control access to other dynamic content, eg PHP dynamic web pages. The Access Control List format was also limited and did not accommodate references to groups of users, which led to large and difficult to manage ACLs. The system was again redesigned and reimplemented in 2003, using the XML-based Grid Access Control Language (GACL) which Manchester had developed and which was already being used by components of the European Data Grid middleware. GACL allowed references to X.509 DNs, groups defined as lists of DNs and VOMS attribute names.

3. mod_gridsite and libgridsite

The other half of 2003's redesign saw the monolithic GridSite CGI replaced with an Apache module, mod_gridsite. All of the code to provide GACL, GSI Proxy and VOMS support, along with utility functions for some HTML and HTTP tasks, was included in libgridsite, a reusable toolkit which can be used by CGI programs hosted on Apache/GridSite, or in standalone client programs or services.

Since mod_gridsite functions are executed within the Apache process, no CGI overhead is added and GridSite can be configured with additional httpd.conf directives. mod_gridsite also dynamically modifies the OpenSSL callbacks used to check X.509 certificate chains, to accept GSI Proxy Certificates and allow GridSite-secured web services to be accessed by running grid jobs and grid services. VOMS attribute certificates are identified whilst the chain is parsed by libgridsite functions and made available for policy enforcement by mod_gridsite, and to CGI programs hosted on the server.

4. Apache/GridSite Architecture



5. Web Services

The modular architecture introduced with mod_gridsite allows the free mixing of static and dynamic content on a single website, whilst retaining the same X.509-based security structure, even for third-party, non-grid systems like MediaWiki and Subversion.

This structure also allowed GridSite to be used as the basis of Web Services within Grid middleware, with the Grid-specific GSI Proxy and VOMS credentials handled by mod_gridsite and passed on to service implementations written as standalone CGI programs or within the FastCGI framework for efficiency.

6. Data Transfer

Apache provides an efficient platform for serving large files, and by adding support for Grid security credentials and the HTTP PUT method, GridSite is able to act as a file server for both reading and writing. Included in the GridSite distribution, the http suite of commands allow scp-style copying of single or multiple files using X.509 or GSI proxy credentials for authentication. Additionally htm and htls allow file deletion and directory browsing using mod_gridsite's support for HTTP DELETE and an extended directory index format.

Experiments over wide area networks have shown comparable performance between GridFTP and HTTPS, but to reduce the CPU overhead of an encrypted data stream, GridSite also provides the GridHTTP profile for using standard HTTP with an unencrypted data stream. A connection is initially negotiated via HTTPS and then an HTTP cookie is obtained and used to retrieve the bulk data file via HTTP. Parallel stream transfers can also be done this way, using Apache's support for partial file retrieval.

7. Cross Site Request Forgery

GridSite's HTTP cookie framework has also been adapted to prevent a class of Cross Site Request Forgery attacks, using the Javascript double-submit cookie method. This stops not only attacks from other websites that target forms and form processing scripts on GridSite-hosted servers, but attacks from one area of the website to another. Website administrators can define zones of increasing privilege and by using a secondary login-only domain name, privilege escalation attacks from one compromised area of a site to another using stolen credentials by CSRF can be prevented using the browser's "same origin policy" and the ability to restrict cookie visibility according to domain name and URL prefix.