The GridSite Toolbar

Shiv Kaushal and Andrew McNab

School of Physics and Astronomy, University of Manchester, Manchester UK

Abstract

We describe the GridSite toolbar, an extension for the Mozilla Firefox web browser, and its interaction with the GridSite delegation web service. A method for automatic discovery of the delegation service is also introduced. The combination of the toolbar and automatic discovery allows users to delegate credentials to a remote server in a simple and intuitive way from within the web browser. Also discussed is the toolbar's ability to enable the use of the GridHTTP protocol, also a part of the GridSite framework, in a similar way.

1. Introduction

Large production grids currently use X.509[1] certificates, GSI[2] proxy certificates and VOMS[3] proxy certificates as a means of authenticating users and delegating authority from users to remote servers. The use of proxy certificates ensures that a user's private key is never exposed but can enable remote servers to act on their behalf.

GridSite[4] is a security middleware project that adds the ability to accept client-side X.509 certificates and GSI/VOMS proxies to the Apache web server[5]. It also allows the generation of access control policies, in GACL[3] or XACML[6], to limit access to files/pages based on these credentials.

GridSite also provides several methods for enabling secure transfer, storage and location of files. One such mechanism, for transferring files over HTTP with access controlled through an initial HTTPS connection, known as GridHTTP, will be explored in the initial part of this paper.

GridSite also acts as a platform for hosting secure web services, written in any of the CGI scripting languages supported by Apache, using the added certificate handling capabilities to authenticate clients. It also comes with a method of “sandboxing” these services inside temporary pool accounts, enabling web servers to safely allow users to remotely deploy web services onto a GridSite server.

As an example web service, the GridSite package contains an implementation of the GridSite delegation service. This is a web service interface for delegation, designed to offer several advantages over the standard methods of transferring proxy certificates to remote servers. The details of the delegation service are discussed later.

A more extensive summary of the GridSite framework can be found in "The GridSite Security Framework"[7].

A common problem when using web services interfaces for applications is knowing how to locate the services. We introduce a method for enabling automatic discovery of a delegation service from within a web browser. The mechanism described could easily be applied to any other web service.

Finally, we describe how this mechanism is used in conjunction with the GridSite toolbar - an extension to the Mozilla Firefox[8] web browser - to allow users to easily locate and delegate credentials to available delegation services. We also show how the toolbar can allow users to make use of the GridHTTP protocol.

2. GridHTTP

GridHTTP is a protocol, defined within the GridSite framework that allows files to be downloaded over a standard HTTP connection, but first requires authentication of the clients via HTTPS. The aim of this protocol is to allow large (gigabyte) files to be transferred at optimal speeds while still maintaining some level of security. The approach used avoids the problem of authentication over HTTP (usually achieved via usernames and passwords) by using the certificate handling capabilities of the GridSite software as well as the access control list functionality.

In order to retrieve a file using the GridHTTP protocol, clients must connect to a web server over HTTPS and set the value of the

"Upgrade" header in the request to "GridHTTP/1.0". This is entirely in keeping with the HTTP standard so any server which does not understand the GridHTTP protocol, or does not have it enabled, will ignore the header.

If GridHTTP is enabled, the server will determine if the client should have read access to the requested files, based on their X.509, GSI proxy or VOMS proxy certificate and any defined access control lists. If access is allowed, the server will respond with a redirection to a standard HTTP location for the file and with a single-use passcode, contained in a standard HTTP cookie. The client should then present this passcode cookie when requesting the file from the HTTP location and will be granted access to download the file.

The performance benefits of the GridHTTP protocol come not only from the data stream being unencrypted (saving CPU cycles at the server and client ends) but also from the highly optimised Apache file serving routines. In comparison, while GridFTP[9] allows unencrypted data transfer it does not make use of low level system calls as in the case of GridHTTP/Apache. The performance difference between the two transfer methods (for unencrypted data streams) is evident in research carried out by R. Hughes-Jones[10].

3. The GridSite Delegation Service

The GridSite delegation service is a web service that allows users to place proxy certificates on remote servers. The WSDL-based service uses standard plain-text SOAP messages, with the authentication coming from an HTTPS connection from the client. This makes it very easy to implement the service in any of a variety of programming languages, especially when using GridSite as the web services platform.

3.1 Delegation Service Internals

The delegation procedure is initiated by a client sending a getProxyReq message to the service. The service then produces a public/private key pair and generates a certificate request from the public key. The certificate request is then sent to the client. The client then signs the request with their private key and sends it back to the server in a putProxy message. The signed request combined with the associated private key forms a valid proxy certificate.

This approach has the distinct advantage over the usual methods of delegating credentials to a remote server. The standard "grid-proxy-init" and job submission routines produce the public and private keys for the proxy certificate locally, sign the public key and then transfer the both to the remote server over the network. In the GridSite delegation service, the private key associated with the proxy certificate never leaves the remote server, adding an extra layer of security to the delegation process.

This is completely analogous to the processes involved in obtaining a standard X.509 certificate, but cast as a web service. The delegation service is treating the client as a Certificate Authority (CA) and requesting that the public key be signed by the user's private key (which acts like the CA's private key). The main difference is that both the client and the service can verify the identity of the other through trusted CAs, so there is no lengthy identification step involved.

The above description covers the functionality of the initial (version 1) delegation service interface. There is now an updated interface specification (version 1.1) that extends the functionality of the delegation service. The new functionality includes methods that allow users to check when an existing proxy certificate will expire, to renew such a certificate and to destroy the certificate.

3.2 GridSite Implementation

The GridSite implementation of the delegation service makes use of the gSOAP[11] toolkit and is written in C. Running under the GridSite environment allows the service to obtain authentication information about connecting clients directly from environment variables. It is intended as a simple illustration of how web services can be created within the GridSite framework but this implementation can also be incorporated into other web services that might require this functionality. Additionally, GridSite provides a command line client, also built using the gSOAP toolkit, in order to supply a complete solution.

The GridSite delegation service specification was created within the EGEE[12] project. As a result, there are also Java client and server implementations available within the gLite framework. Since the service is based on a WSDL definition, all of the combinations of client and server implementations interoperate without any problems.

4. Service Discovery

Traditional web services require that the client is either configured for one particular instance of the service, with a hard-coded location (as is the case with the GridSite delegation service command line client, once compiled), or that the user of the client provides the URL of the service they wish to use. This can cause problems if locations of services change or users wish to locate alternative services providing the same functionality.

A system was developed to allow a browser (and in turn the GridSite toolbar) to be notified of an available delegation service by a web site. The URL of the service is provided either in HTTP headers or in a META[13] element in the HTML source of the page being viewed. The META element method is similar to the method employed by sites and browsers to enable automatic location of RSS feeds, which uses a LINK element.

The header used for the notification is "Proxy-Delegation-Service", the value of which should be the URL of the delegation service. The insertion of this header can be easily achieved with GridSite Apache module and setting the GridSiteDelegationURI directive in the web server's configuration file.

To deliver the same information without using HTTP headers, a META element containing the http-equiv attribute can be inserted into the HEAD element of a web page. The META element would have the following format:

```
<META
http-equiv =
"Proxy-Delegation-Service"
content =
"https://eg.com/delegate.cgi"
>
```

The combination of the two methods above provides great flexibility for different groups of people to alert users to a delegation service. The HTTP header method allows a web server administrator to inform all visitors of a related delegation service and META element method allows an individual page (a personal home page, for example) to do the same. The latter may be useful in an environment where enabling server-wide options are not possible (e.g. pages located on unrelated servers) or on a server not using the GridSite software.

5. The GridSite Toolbar

As stated previously, it is possible for users to upload custom web services to a GridSite server, an example of which is the delegation service. Additionally, GridSite provides a wide variety of site management features, including editing/uploading/deleting files and folders and editing access control lists. All of these tasks can be carried out from within a web browser.

Although a command line client was created for the delegation service, it did not provide an easily accessible interface as in the case of the browser enabled functionality mentioned above. A browser based client for the delegation service was produced to illustrate a mechanism for allowing interaction with such web services.

The GridSite toolbar is a client for both the GridSite delegation service and the GridHTTP protocol, wrapped up in an extension for the Mozilla Firefox web browser. It makes use of several features of the browser and the service discovery method, described in Section 4, to make delegation and using GridHTTP a simple point-and-click task.

5.1 Mozilla Firefox

Mozilla Firefox was chosen as a base platform for the GridSite toolbar for several key features. These include:

- Default web browser in the Scientific Linux distribution, recommended by EGEE/gLite.
- Explicitly designed to be easily extensible.
- Provides JavaScript objects for manipulation of SOAP messages and interacting with WSDL services.
- Provides a cross-platform development environment.
- Simple API for creating graphical interface elements.
- Built in certificate verification of remote servers over all HTTPS connections.
- Security updates come "for free" from Mozilla.

In addition to these features, the use of Firefox keeps to the GridSite project’s general philosophy of building on established, open source software and related protocols, such as Apache and HTTP, as much as possible. These projects have large development teams and are extensively tested by a wide user base. This allows the GridSite software to inherit the stability, performance and security of these projects and receive security updates for crucial elements (such as HTTPS communication – client and server side) for free. Trying to create stable, efficient solutions for the functions that GridSite and the toolbar provide in the form of bespoke software and protocols would be much more time consuming and much harder to maintain against possible security flaws.

5.2 Requirements

The GridSite toolbar makes several assumptions about the configuration of a user's environment. It requires that the user's certificate (as well as the relevant CA root certificate) is loaded into the Firefox software security device. It is also assumed that the user’s certificate is stored in PEM encoded usercert.pem and userkey.pem files in the user's ~/.globus directory. Finally,

the extension also requires OpenSSL[14] command line tools to be installed.

The toolbar has been developed to work on Linux systems but could be ported to work in a Windows environment, provided that a method of producing secure named pipes, or an equivalent, is available (see section 5.4 for details of how these are used).

5.3 Service Detection

Every time a page (or tab) is loaded or brought into focus, the HTML source of the displayed page is searched for the relevant META element. Then an HTTP HEAD request is made to the same URL as the page being displayed and the response is inspected for the Proxy-Delegation-Service header. If location is defined in both the HTTP headers and the META element the value found in the headers will be used.

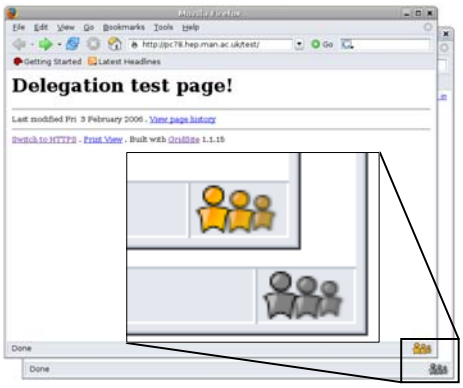

Figure 1: The two states of the delegation service detection status bar icon

The detection of a delegation service is indicated by an icon in the status bar of the browser window (as shown in Figure 1). When a service is detected, the icon is "lit up" and becomes active. In this state, clicking it will initiate the delegation procedure. When no delegation service is detected, the icon will become greyed out and clicking it will produce a message informing the user that no service was found.

5.4 Delegation Procedure

The delegation procedure involves several steps. Firstly a getProxyReq SOAP message is constructed and sent to the discovered URL. The service responds with a certificate request, which is saved to disk in a temporary location.

This step (and every subsequent communication with the delegation service) is carried out using Firefox's HTTPS capabilities. The browser's built in functionality will produce a warning to the user if the server hosting the delegation service has a certificate that is either invalid or not from a recognised CA. The connection also uses the certificate in the software security device to authenticate to the service.

The user is then asked for their PEM passphrase, which is passed, through a named pipe, to a shell script wrapped around the OpenSSL command line program. This script signs the request to produce the proxy certificate. The shell script makes use of a file called usercert.srl in the user's ~/.globus directory (from the -CAcreateserial command line option for OpenSSL X.509 tools) to ensure that the extension will never produce two proxy certificates with the same serial number – as recommended in the IETF RFC3280 specification for GSI proxies.

Finally, the certificate is read in, inserted into a putProxy SOAP message and sent to the delegation service.

5.5 User's Experience of Delegation

The majority of the details described above are hidden from the user when using the toolbar. Upon clicking the delegation button in the status bar, a short dialogue box appears informing the user that the browser is attempting to connect to the delegation service.

The user may then be prompted for up to 2 passwords. The first is for the Firefox software security device and will only be asked once per Firefox session. This is the standard Firefox behaviour when using the user's certificate to authenticate to a secure server. The second password requested is the PEM passphrase for the ~/.globus/userkey.pem file. For security purposes, this passphrase is requested every time the GridSite toolbar is used to delegate to a service.

Once the proxy request has been signed and sent back to the server, a confirmation dialogue box will appear.

5.6 Limitations

There are some limitations in the current version of the GridSite toolbar with regards to the interaction with the GridSite delegation service. Firstly, proxy certificates can only be created with validity times in multiples of 24 hours. This is due to restrictions in the OpenSSL command line tool. The time is currently locked at 24 hours, but there is room to allow for varying lifetimes.

Additionally, there is currently no method offered to allow users to select a service from

META tags in preference over a service specified in HTTP headers. This can be achieved through the use of directory specific Apache settings by disabling the HTTP header for a particular area of a site.

Finally, the new features introduced in version 1.1 of the delegation service interface are not yet supported by the GridSite toolbar. This could be achieved relatively easily by extending the work already done.

5.7 GridHTTP

Access to files using the GridHTTP protocol is similarly an easy operation when using the GridSite toolbar. To do so, users right click a HTTPS link to a file and select "Get with GridHTTP" (Figure 2).

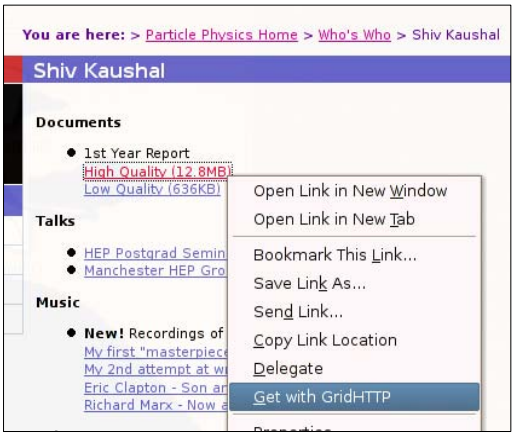

Figure 2: Using GridHTTP to download files from within Firefox.

The extension then sets the location of the current window to the right-clicked URL and intercepts the request to add the required Upgrade header. After this point no further intervention is required from the toolbar. The default Firefox behaviour is to follow the redirection from the server and to present the passcode cookie. Upon doing this the browser will handle the file as normal – either displaying it within the browser window or prompting the user to save or open the file with an external application.

6. Conclusion

The GridSite toolbar is a simple example of how the GridSite functionality can be combined with the service discovery method and a web browser environment in order to simplify the use of web services. It not only avoids the use of long command line strings, but also uses the built in functionality of the web browser (such as choosing where to save a file) to add graphical elements that make the process more intuitive.

Using the web service hosting features of GridSite and the methods outlined in this paper, it is possible for any user to create and deploy a web service, or a collection of services, configured to their own specific set of requirements. The web service can be written in almost any programming or scripting language, hosted as CGI scripts for Apache, and then made accessible through the familiar environment of a web browser.

Applications could range from something as simple as a having a mechanism for notifying users of the status of submitted jobs to a complex Grid "portal" site, which could be used to submit jobs and retrieve output. In the latter case, the GridSite delegation service and GridSite toolbar functionality could be easily integrated to allow the delegation of credentials to the service as required.

The GridSite toolbar demonstrates that making grid applications as easy to use as any locally installed application is possible. Using such techniques can help make the Grid more accessible for new users and ease its adoption.

Acknowledgements

This work was funded by the Particle Physics and Astronomy Research Council through their GridPP end e-Science Studentship programmes.

We would also like to thank other members of the EDG and EGEE security working groups for providing much of the wider environment into which this work fits.

References

1. X.509v3 is described in IETF RFC2459, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile."
2. Grid Security Infrastructure information is available from Globus: http://www.globus.org/toolkit/docs/4.0/security/
3. VOMS and GACL are described in the EDG Security Co-ordinations Group Paper, "Authentication and Authorization Mechanisms for Multi-Domain Grid Environments", L. A. Cornwall et al, Journal of Grid Computing (2004) 2: 301-311.
4. GridSite Software is available from http://www.gridsite.org/
5. The Apache Web Server: http://httpd.apache.org/
6. XACML specification is by OASIS: http://www.oasis-open.org
7. "The GridSite Security Framework", A. McNab & S. Kaushal, Proceedings of All Hands Meeting (2005).
8. Mozilla Firefox information and downloads: http://www.mozilla.com
9. GridFTP: http://www.globus.org/grid_software/data/gridftp.php
10. Richard Hughes-Jones, private communication and talk at GNEW 2004.
11. The gSOAP C++ web services toolkit is available from http://www.cs.fsu.edu/~engelen/soap.html
12. The EGEE (Enabling Grids for E-sciencE) project: http://public.eu-egee.org/
13. HTML 4.01 Specification detailing the use of all valid elements: http://www.w3.org/TR/REC-html40/
14. OpenSSL is available from http://www.openssl.org/