Communicating Security Assertions over the GridFTP Control Channel

Rajkumar Kettimuthu[1,2], Liu Wantao[3,4], Frank Siebenlist[1,2], and Ian Foster[1,2,3]

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL USA
[2]Computation Institute, The University of Chicago, Chicago, IL USA
[3]Department of Computer Science, The University of Chicago, Chicago, IL USA
[4]Beihang University, Beijing, China

The GridFTP [1] protocol defines a general-purpose mechanism for secure, reliable, high-performance data movement. GridFTP has been widely used for efficiently transferring large volumes of data. It is based on the Internet FTP protocol and thus involves two communication channels: a control channel and a data channel. The commands and responses flow over the control channel, and the data is transmitted over the data channel.

The Globus implementation of GridFTP [2] has a modular structure that supports multiple security options, multiple transport protocols, coordinated data transfer using multiple computer nodes at the source and destination, and other desirable features. The Globus GridFTP design supports secure authentication of control channel requests via the Grid Security Infrastructure (GSI), Kerberos, or an SSH security mechanism. GSI is the commonly used security mechanism for GridFTP transfers.

In this work, we develop a mechanism to reduce the security overhead in authenticating and authorizing the users to perform GridFTP transfers in portal environments. We describe how a control channel is established using GSI security mechanism. Specifically, the client initiates a TCP connection to the port on which the server is listening. Needed first is authentication via RFC 2228. By default, the client presents a delegated proxy certificate [3], and the server must present a host (or user) certificate issued by a certificate authority trusted by the client. If authentication is not successful, the connection is dropped. If authentication is successful, an authorization callout is invoked to verify authorization and determine the local user id with which the request should be executed. This callout is linked dynamically. Typically, the local user id is obtained from a Globus gridmapfile, which contains a mapping of Distinguished Name (DN) in user's certificate to local user ids. The server does a setuid to the local user id as determined by the authorization callout. If authorization succeeds, the control channel is established and the rest of the control channel protocol exchange can proceed.

In environments with a large number of users, services such as Community Authorization Service (CAS) [4] and Virtual Organization Management Service (VOMS) [5] have been developed to address the scalability issues with the Globus gridmapfile approach. These services enable multiple users to have the same DN and encode in Security Assertion Markup Language (SAML) [6] assertions (embedded as extensions in proxy certificate) the specific files that a user is authorized to read and/or write. These services maintain the permissions of users in a virtual organization; the individual sites do not need to have a large number of user accounts and/or maintain long gridmapfiles.

Consider a web portal where multiple users logon and initiate third-party data transfers between two remote nodes. It is possible that more than one user want to move data between the same pair of sites. Each user either has his own certificate or gets a community certificate from a service such as CAS or VOMS that has his permissions embedded as a SAML assertion. Either way each user's certificate is different, and a separate control channel needs to be established for each user's transfer request.

If a separate control channel is needed for each user, it is difficult for the portal to cache the control channels and reuse it. The Earth Systems Grid [7] project, for example, needs to transfer many small files and will have a separate SAML authorization decision assertion for each of those transfer requests. The percentage overhead associated with the initiation of a GridFTP session (or control channel) could be significant when the size of the file to be transferred is small. We develop enhancements to GridFTP to avoid the overhead by reusing a single control channel for multiple file transfer operations (from the one or more users). The portal would use a single proxy certificate for all the users. Currently, the SAML assertions are embedded in the proxy certificate that is used by the client to authenticate to the GridFTP server. The objective is to provide the GridFTP clients with the ability to specify a SAML-assertion per GridFTP data transfer command while reusing the existing established session between the client and the GridFTP server.

The proposed solution is to use the GridFTP SITE command to let the client communicate a SAML assertion to the GridFTP server where it will be used for the next authorization decision in the authorization call-out. Any subsequent SITE directive that communicates a new SAML assertion will substitute and therefore override the previous one, which will allow the next GridFTP commands to use the last SAML assertion that was communicated.

A new command SITE AUTHZ_ASSERT has been added to the Globus GridFTP framework. A new API has been added to the Globus FTP client library that allows the passing of SAML assertion to the GridFTP server. The commonly used GridFTP command line client 'globus-url-copy' has also been updated to allow the users to pass different SAML assertions for each transfer URL. SAML assertion will be presented to the GridFTP client as a string and it will not have to interpret it. For third party transfers, clients may have to send different security assertions to the source and destination. Support for sending different assertions to source and destination GridFTP servers has also been added.

The GridFTP server has been updated to support the new SITE command. The server adds the received SAML assertion to the security context such that it can be picked up by the authorization call-out. An API that takes the SAML assertion string and an exiting security context, and adds assertion to the context, has been added.

On the authorization call-out side, the implementation has been changed to look for the SAML blob that came in over the control channel and was passed inside of the security context. If an assertion is found, it is used in the identical way the call-out handled the assertion in the proxy certificate before. If the call-out cannot find a control-channel SAML blob, it looks for the SAML blob in the proxy certificate. Preliminary results show that significant improvements in performance – up to multiple orders of magnitude – can be achieved with this enhancement.

Acknowledgment

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

References

[1] Allcock, W. GridFTP: Protocol Extensions to FTP for the Grid. Global Grid ForumGFD-R-P.020, 2003.

[2] Allcock, W., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I., and Foster, I., The Globus Striped GridFTP Framework and Server, in SC'05, ACM Press, 2005

[3] Tuecke, S., Welch, V., Engert, D., Pearlman, L., and Thompson, M. Internet X.509 Public Key Infrastructure Proxy Certificate Profile. Internet Engineering Task Force, RFC 3820, 2004.

[4] Pearlman, L., Welch, V., Foster, I., Kesselman, C. and Tuecke, S., A Community Authorization Service for Group Collaboration. IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[5]https://twiki.cnaf.infn.it/twiki/view/VOMS/

[6] http://xml.coverpages.org/saml.html

[7] http://www.earthsystemgrid.org/