Appears in the Proceedings of the 2005 International Conference on Computational Science (ICCS 2005), Emory University, Atlanta, GA, May 22-25, 2005.

Extending Existing Campus Trust Relationships to the Grid through the Integration of Pubcookie and MyProxy*

Jonathan Martin[1], Jim Basney[2], and Marty Humphrey[1]

[1]Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA
[2]National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign, Champaign, IL, 61820, USA

Abstract. In many ways, the ultimate success of the Grid will be highly dependent on the support for and integration with existing legacy infrastructure such as authentication infrastructure. This paper describes the design and implementation of the integration of Pubcookie, a popular Web-based authentication infrastructure used on campuses, and MyProxy, the on-line credential repository for the Grid. Specifically, we enable a valid Pubcookie credential to be dynamically exchanged for a Grid credential, without requiring the user to re-authenticate. Through this integration, this project makes an important contribution to an overall goal of single sign-on and more specifically the ability to "authenticate locally and act globally".

1 Introduction

Through a largely global effort, the last few years have seen a significant expansion and hardening of Grid technologies, particularly the maturation of the Globus Toolkit [1] and related technologies. Although the underlying software of the Grid can always be improved -- particularly the ability of the software to tolerate faults -- scientists are increasingly relying on Grid software to enable and manage their scientific explorations. The next version of the Globus Toolkit, based on WSRF [2], along with the WSRF-compliant hosting environment for the .NET Framework (WSRF.NET [3][4]), promises to expand these capabilities even further by creating uniform mechanism across Linux/UNIX and Windows, respectively.

However, one of the continuing challenges of the Grid software is to accommodate *legacy* mechanisms and policies. To make an existing scientific application "grid-aware", one must address issues related to security, I/O, scalability, licensing, etc., that often require non-trivial modifications. To truly be successful and a ubiquitous global computing infrastructure, the Grid must require fewer modifications to existing behaviors, particularly the manner in which scientists are accustomed to interacting with local resources. When the local scientist *believes* she is performing experiments and

* J. Martin and M. Humphrey are supported in part by the US National Science Foundation under grants ACI-0203960 (Next Generation Software program), SCI-0438263 (NSF Middleware Initiative), and a subcontract from the San Diego Supercomputing Center (SDSC).

manipulating data on a local set of resources when she is *actually* using non-local Grid resources, the Grid will be an unequivocal success.

This is particularly true with regard to security -- existing security policies, mechanisms, and trust relationships must be leveraged instead of being displaced by the Grid. For example, assume that a campus researcher has previously set up a secure Web portal for access either to local data or local compute cycles. To protect her resource, she has used Pubcookie [5], a popular open-source package for intra-institutional single-sign-on end-user Web authentication. Recently, she has been overwhelmed by local and non-local requests for this content and is investigating using the Grid (such as the TeraGrid [6] or the Open Science Grid [7]) as a back-end compute engine and data store as needed. However, she wants the back-end use of the Grid resource to be as seamless as possible and, unfortunately, she cannot afford to pay for the Grid resource usage on behalf of the requestors. Instead, these Grid computations and data request/storage must be performed by (and charged to) the requestor himself. In this situation, it is unacceptable to require the scientist to replace her existing local authentication with the Grid Security Infrastructure (GSI [8]), which is required of most Grid resources.

This paper describes our project to re-use existing campus trust relationships and authentication infrastructures when dynamically expanding to Grid computing resources on demand as in the scenario above. We bridge the campus environment of Pubcookie with the Grid environment of GSI through the MyProxy on-line credential repository [9][10]. In essence, the possession of a valid Pubcookie token is used as the basis for retrieving a Grid credential from the MyProxy server. In doing so, the user does not have to re-authenticate to the Grid -- the Pubcookie credential is *exchanged* for a valid credential to be used on the Grid. More broadly, this project makes an important contribution to an overall goal of single sign-on and more specifically the ability to "authenticate locally and act globally".

This paper is organized as follows. In Section 2, we present the existing projects that are related to this work. In Section 3, we describe Pubcookie and MyProxy in more detail and describe the integration design and implementation. Section 4 concludes and describes the future direction of this project.

2 Related Work

In this section, we describe the related work -- the Grid technologies in Section 2.1 and then the non-Grid technologies in Section 2.2. MyProxy and Pubcookie are presented separately in Section 3, along with the design for the integration of the two technologies.

2.1 Grid Technologies

There are a number of excellent projects that contribute pieces toward overall Grid security today. The Grid Security Infrastructure (GSI [8]) focuses on an authentication infrastructure for the Grid that is based on a Public-Key Infrastructure (PKI). GSI

provides a standard programming interface for authentication, message integrity, and message confidentiality (GSSAPI), a mutual-authentication mechanism based on SSL/TLS, and a delegation protocol by which a user can temporarily empower a software service to act on her behalf. GSI also supports restricted delegation via proxy certificates; however, to our knowledge, to date, this capability is not used in most situations. The Community Authorization Service (CAS [11]) extends the base support of GSI so that a person can obtain and exercise authorization rights based on the group to which they belong. Additionally, two projects focus on the generation and creation of the *gridmap* file, which is used to both authorize users and specific the local account to which the global (grid) account is to be mapped: VOMS [12] manages a list of "Virtual Organizations" that define this mapping, while Walden [13] retrieves this list from a local authorization source such as an LDAP server.

2.2 Campus/Enterprise Technologies

Internet2 is leading the development of many important middleware projects aimed at the campus environment, focusing on directories and PKI. Shibboleth [14] is an open-source, privacy-preserving federating software project to support inter-institutional sharing of web resources subject to access controls. Essentially, when a user at one institution (the “Origin”) tries to use a resource at another (the “Target”), Shibboleth sends attributes about the user to the Target institution without having to log into the target institution. The Target institution grants access based on the attributes. The user controls what attributes are given to the Target institution (for example, it is not strictly necessary that the attributes include the name of the user, if the Target institution bases its decision on, say, “member of UVa”). Stanford’s Signet [15] is software to define and manage an organization’s privilege system, with special emphasis on how to take a role-based organization and develop appropriate groups, policies, and attributes to operate an authority service. Signet is *not* an authorization service, but rather integrates with authorization services. An example use of Signet is a Web-based interface for assigning workers on campus the ability to make constrained purchasing decisions for a limited period of time. Grouper [16] offers support for basic group management, with subgroups and compound groups. To date, none of these projects have been made Grid-aware in the manner in which we have integrated Pubcookie with MyProxy as described in this paper (an effort to make Shibboleth Grid-aware has just commenced as of this writing).

3 Integration Design and Implementation

In this section, we first give the details of Pubcookie, and then MyProxy, and then we give the design for how we integrate Pubcookie and MyProxy to effectively connect the campus to the Grid without requiring a second sign-on.

3.1 Pubcookie

As shown in Figure 1, Pubcookie [5] provides single sign-on authentication to web sites using existing site-wide authentication services. In this way, a user who has already authenticated for access to one web site can access other protected sites without having to enter another password. For example, a user who has authenticated to a web mail system could browse a separate web database system without needing to log in again.

Pubcookie provides the glue between the authentication service and the web site and is not responsible for the actual authentication or authorization of users. Pubcookie does not verify who the user is – it hands this task off to a separate authentication service (like Kerberos [17], LDAP, or NIS). Likewise, Pubcookie does not decide if a user should be allowed to access a given page or resource. Control of access to the resource is left to the originating application.

This is accomplished, as the name implies, with cookies. When a user initially attempts to access a protected site (Arrow #1 of Figure 1), the Pubcookie module at that site automatically redirects her browser to a Pubcookie login page with a "granting request" cookie containing the request (the URL) and a random number (#2). The user enters her username and password on the login page (#3), which Pubcookie verifies using the configured authentication service (#4 and #5). Pubcookie then returns two cookies to the user's browser (#6) and redirects the user back to the original site (#7).

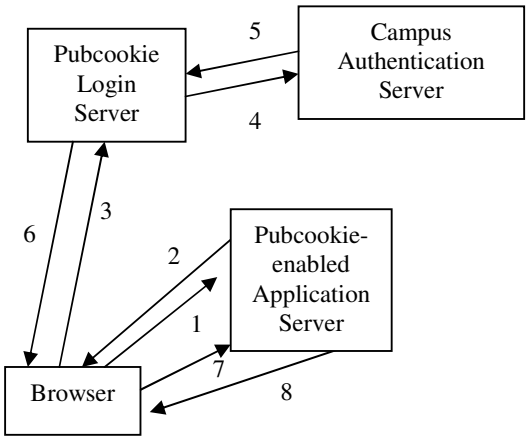

Figure 1: Pubcookie Usage (without MyProxy integration)

The first of these cookies returned in step #6 of Figure 1 is a "granting cookie". This cookie authenticates the user to the target web site ("Pubcookie-enabled Application Server"). The cookie contains the information in the "granting request" cookie

plus the authenticated username. When the Pubcookie login server redirects the browser back to the original web site (#7), the Pubcookie module at the web site verifies the "granting cookie" and determines if the user is authorized to access the web page. If the user is authorized, the web server returns the requested content (i.e., the web page) with a session cookie to authenticate subsequent requests at that site (#8). The second cookie returned by the login server is the Pubcookie "session cookie". As its name implies, this is the cookie that the user maintains throughout their Pubcookie session. When the user visits a new Pubcookie-enabled web site, it will again redirect her to the Pubcookie login server. However, now that the user has a Pubcookie "session cookie", she does not have to login again (Steps #4 and #5). The login server verifies the session cookie and automatically redirects the user back to the new web site with a new "granting cookie". Thus, the user has a single sign-on to all Pubcookie-enabled web sites.

There are two means by which the security of these cookies is guaranteed. First, they are encrypted. When Pubcookie is being set up, the Pubcookie login server (which issues the cookies) and the application server (i.e., the web server the user is trying to access) exchange a cryptographic key. This key is in turn used to encrypt and decrypt the data used by Pubcookie. The second method by which security is achieved is digital signatures. The Pubcookie login server digitally signs all cookies it issues, so that application servers can use the login server's public key to verify the authenticity of the cookies it issues.

3.2 MyProxy

MyProxy [9][10] is a service for securely storing GSI credentials. MyProxy uses the proxy delegation protocol to allow clients to retrieve short-lived proxy credentials without exporting long-lived keys from the MyProxy server. A dedicated MyProxy server provides more secure storage for user keys than a general-purpose workstation or file-server, and MyProxy can be integrated with cryptographic hardware to further protect user keys [18]. To retrieve a proxy credential from the MyProxy repository, the MyProxy client must first authenticate. Current versions of MyProxy support authentication via password, certificate, Kerberos, or PAM (Pluggable Authentication Modules). This allows users to retrieve proxy credentials from the MyProxy server whenever and wherever they are needed. Additionally, MyProxy can be used to renew credentials for trusted long-lived services.

MyProxy is widely used with grid portals, which provide a web interface to grid services. To allow users to interact with secure grid services through the portal interface, the portal must have access to the user's grid credentials, so it can perform secure operations on the user's behalf. To meet this requirement, users allow the grid portal to retrieve their credentials from the MyProxy server, by sending their MyProxy username and password to the portal, which it then uses to authenticate to MyProxy and retrieve the short-lived credentials. Given the security issues associated with web server platforms, this limits the vulnerability of user credentials on the portal by giving the portal access to credentials with a limited lifetime, with access logged at the

MyProxy server, with the user able to change her MyProxy password at any time to deny the portal any further access to the credentials.

3.3 Integration

Because Pubcookie successfully implements single sign-on for the campus/enterprise environment, and MyProxy/GSI successfully implements single sign-on for the Grid environment, it is highly attractive to investigate the design, cost of implementation, and the implied security of using one credential as the basis for acquiring the *other* type of credential (i.e, Pubcookie-for-MyProxy or MyProxy-for-Pubcookie). Because our specific goal was to not disrupt the local campus environment in attempting to provide the ability to dynamically expand the local resources to a Grid such as the TeraGrid, we chose to use a valid Pubcookie as the basis for retrieving a GSI credential from MyProxy. More specifically, the user authenticates to the grid portal using the Pubcookie mechanism and the grid portal retrieve GSI credentials for the user from MyProxy by authenticating with the Pubcookie "granting cookie" without requiring the user to enter an additional password (for MyProxy). As shown in Figure 2, the system appears to the end-user as before (without MyProxy integration) but now has the option to expand onto the Grid, perhaps even without the local user realizing this is being done (Steps #8 through #11).

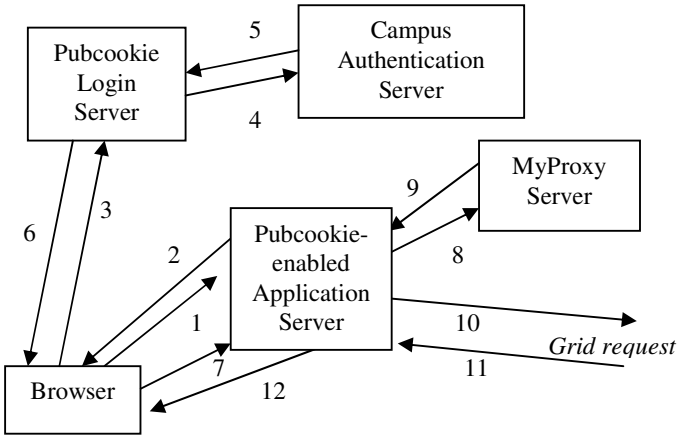

Figure 2: Pubcookie Usage (with MyProxy integration)

To achieve this, we modified the Pubcookie-enabled Application Server software to send the signed authentication data from the decrypted cookie to MyProxy (#8), over an SSL authenticated and encrypted channel. The SSL channel is mutually authenticated, so the application server knows that it is communicating with the trusted MyProxy server, and the MyProxy server knows the identity of the application server, to be compared against the server name in the cookie, to ensure the cookie has not been stolen. For added security, the MyProxy server can be configured to only allow

connections from specific trusted application servers (such as the specific Pubcookie-enabled Application Service shown in Figure 12). The MyProxy server then verifies the cookie's signature, verifies that the username in the cookie matches the MyProxy username, verifies that the server name in the cookie matches the portal's authenticated SSL name, and if verification succeeds, delegates the proxy credentials (#9). MyProxy uses the OpenSSL library to verify the signature; no Pubcookie code was added to the MyProxy software.

Our development was based on the widely-available source code for MyProxy from the MyProxy web page [10]. These modifications are being made as permanent additions to the MyProxy source code. The tests and verification were performed at the University of Virginia's Pubcookie installation, which is a modified version of Pubcookie 3.1.1. The modifications include the ability to use Radius authentication as the connection to the local authentication source and also SMB authentication.

It is important for the MyProxy username to match the Pubcookie username, so the portal uses the correct GSI credentials for each Pubcookie authenticated user. Multiple mechanisms are available to ensure this. First, the site's security administrators may load the MyProxy repository with user credentials under the correct usernames. This configuration relies on the MyProxy server, Pubcookie authentication service, and site Certification Authority (which creates the GSI credentials) being under the same administrative control. A second option is for the MyProxy administrator to maintain an access control list mapping Pubcookie usernames to MyProxy usernames. The mapping could support multiple Pubcookie servers by formatting Pubcookie usernames as *user@server*. The third option is for MyProxy users to set the Pubcookie authorization policy for their credentials when they upload them to the MyProxy repository, using MyProxy's existing per-credential authorization functionality. The MyProxy administrator must also configure the server to trust the Pubcookie login server signing key(s).

4 Conclusion

In many ways, the ultimate success of the Grid will be highly dependent on the support for and integration with existing legacy infrastructure. In this paper, we described our support for re-using existing campus/enterprise authentication infrastructure in combination with the Grid. That is, we have successfully and securely implemented modifications to MyProxy so that a valid Pubcookie-issued cookie could be used as the basis for acquiring a GSI credential.

Having successfully completed the modifications to MyProxy, we are currently rewriting our Grid portal that we have developed at the University of Virginia as part of the National Partnership for Advanced Computational Infrastructure (NPACI). The purpose of this portal is to provide an easy-to-use web interface for Computational Biophysics (focusing on applications such as CHARMM, Amber, and NAMD) [19]. This portal is currently protected via usernames and passwords. By making this portal protected by Pubcookie (while still supporting username/password when a user does not have the ability to be authenticated via Pubcookie), and making the portal connect

to a Pubcookie-speaking Myproxy server, we significantly increase the ease-of-use for the computational scientist.

References

[1] Globus project, www.globus.org

[2] K. Czajkowski., Ferguson, D., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenepe, W. 2004. The WS-Resource Framework. http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf

[3] WSRF.NET: The Web Services Resource Framework on the .NET Framework. http://www.ws-rf.net

[4] Humphrey, M., G. Wasson, M. Morgan, and N. Beekwilder (2004). An Early Evaluation of WSRF and WS-Notification via WSRF.NET. *2004 Grid Computing Workshop (associated with Supercomputing 2004).* Nov 8 2004, Pittsburgh, PA.

[5] Pubcookie project. www.Pubcookie.org

[6] TeraGrid project, www.teragrid.org

[7] Open Science Grid project, www.opensciencegrid.org

[8] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pg. 83-92, 1998.

[9] J. Novotny, S. Tuecke, and V. Welch. An Online Credential Repository for the Grid: MyProxy. Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[10] MyProxy on-line Credential Repository Project.

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

[12] Virtual Organization Membership Service (VOMS). http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html

[13] B. Kirschner, T. Hacker, W. Adamson, B. Athey. Walden: A Scalable Solution for Grid Account Management, *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. November 8, 2004. Pittsburgh, PA.

[14] Shibboleth. http://shibboleth.internet2.edu

[15] Internet2 MACE – Signet. http://middleware.internet2.edu/signet/

[16] T. Barton and B. Christiansen, eds. Grouper Phase 1 Specifications. Draft of 3 May 2004. Available at: http://home.uchicago.edu/~tbarton/draft-barton-christensen-grouper-phase1-specs-04.html

[17] B.C. Neuman, and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33-38, September 1994.

[18] M. Lorch, J. Basney, and D. Kafura, "A Hardware-secured Credential Repository for Grid PKIs," 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2004), Chicago, Illinois, April 19-22, 2004.

[19] NPACI Computational Biophysics Portal. https://wumpus.cs.virginia.edu/NPACIComputationalBiophysicsPortal/