Authenticating from multiple authentication sources in a collaborative platform

Quang Vu DANG[1], Olivier BERGER[1], Christian BAC[1], Benoît HAMET[2]
1 Groupe des Écoles de Télécommunications-Institut National des Télécommunications,
9 rue Charles Fourier 91070 Évry France
{Vu.Dang-Quang,ChristianBac,Olivier.Berger}@int-evry.fr
WWW home page: http://proget.int-ery.fr/projects/PFTCR/
2 PhpGroupWare Project
Benoit.Hamet@laposte.net,
WWW home page: http://www.phpgroupware.org

Abstract. This paper presents a proposal to address the need for multiple authentication sources for users of collaborative work platforms. The proposed approach, developed for the needs of GET and Picolibre, relies on a generic solution that integrate groupware servers in a Shibboleth infrastructure. We have developed adapters for this integration, that we contributed to the phpGroupware project. This document should serve as a basis for discussion in order to validate the level of generality of the proposed approach. We hope that this approach can also help maintainers of other collaboration platforms, who want to integrate a park of deployed platforms with external user identification and authentication services, get a better view of solutions available with Shibboleth.

1 Introduction

The Groupe des Écoles des Télécommunications (GET) is composed of several engineering and business schools together with research centres in Paris (ENST), Brest (ENST Bretagne) and Évry (INT), in France. The research teams are made up of more than 600 full-time research equivalents. The range of the researchers' expertise, from technologies to social sciences, enables the integrated approach so characteristic of GET research and fosters its adaptability to new application sectors and new usages in response to current challenges in the field of Information and Communication.

Starting in order for use as a pedagogical platform, Picolibre [1] is a libre-software system developed at GET, and released under the GNU GPL license. It

Please use the following format when citing this chapter:

Dang, Q.V., Berger, O., Bac, C., Hamet, B., 2007, in IFIP International Federation for Information Processing, Volume 234, Open Source Development, Adoption and Innovation, eds. J. Feller, Fitzgerald, B., Scacchi, W., Sillitti, A., (Boston: Springer), pp. 229–234.

provides a Web-based collaborative work platform built on top of phpGroupware[1] and other libre software tools. Picolibre provides project hosting facilities for small teams of software developers, which was mainly oriented to teaching and research environments[2].

Picolibre integrates several libre software Web applications, but lacked some features in the current stable version, like a wiki engine. However, we have developed an in-house GET platform, based on Picolibre components, integrating new services like a Wiki engine, or a Webdav server, called ProGET [2]. We have integrated these new features in a new generation PicoLibre that is called PicoForge. This new generation is a more generic and complete solution, and available for all as libre software. Several platforms have been deployed at GET, and developers or researchers may be using services of several such platforms, while working on projects initiated on these different platforms.

At the present time, different accounts may be created on these platforms for the same person. Here comes the need of Single-Sign-On (SSO) facilities between these platforms. GET is in the process of deploying soon a federation of authentication systems and applications, based on Shibboleth, for a better integration of its Information System, which is at the present time distributed among the different schools.

Shibboleth is an infrastructure designed to build an identity federation allowing applications and identity providers to share and exchange attributes concerning user profiles, in order to facilitate user identification and authentication in the realm of a deployed identity federation. We investigated the use of that infrastructure, but we should note that users of our platforms may be either already registered in GET's “company directories”, or external contributors unknown to these directories. Our platforms were not interfaced with our company directories to authenticate its users, and manages its own directory. Here comes the need to develop adapters to integrate Picolibre platforms in the coming Shibboleth federation. But even if company users are recognised by Picolibre, through its use of Shibboleth, we still have to support other users not known of companies directories. Such requirement will also be described in our proposed solution.

The same issues, which are addressed here for Picolibre users and GET, may be found also for other networks of collaborative work platforms, for instance among libre software development communities, for authentication into the various software development platforms they use (Gforge, Trac, etc.).

[1] http://www.phpgroupware.org/

[2] The reader will find more details about PicoLibre at picolibre.org/

2 Shibboleth and SSO Service

Shibboleth[3] is a complete open source platform developed for project "Internet2" [4], aiming at building the federation of identity for education institutions and their partners. It is based on the SAML standard (Security Assertion Markup Language), which defines the assertion of authentication and exchange of attributes of users. It supports the SSO service and the authorization, allowing the definition of access control privileges to Web resources, by using user-associated attributes.

The basic architecture of Shibboleth has three components: Identity Providers (IdP), Service Providers (SP), and a "Where Are You From" service (WAYF). **Identify Providers** (IdP) are responsible of user authentication, and to provide the user attributes for the access control process. **Service Provider** (SP) is managing access to the resources. The Apache web server "plugin" *mod_shib* is a module that allows Web pages access control, based on the attributes values defined in the IdP. The service **"Where Are You From"** (WAYF) helps the user to explicitly choose his/her IdP “of origin”, selecting the place where he/she may be known as a source of authentication.

When a user wants to access a SP, it is redirected towards an IdP, or a WAYF service for choosing the IdP. Then the user authenticates to this IdP. After successful authentication, the user is directed back to the requested service, but being authenticated, this time. The SP also requires informations that describes the user, and filters these informations for the authorization process. These informations are sent to the service's Web applications in HTTP headers or cookies.

3 Enhanced authentication in Picolibre

At the present time, users authenticate against the phpGroupware application, which provides the basis for users management in PicoLibre. They log-in when they want to access the “virtual desktop” homepage of PicoLibre, which contains the list of the projects to which they collaborate. User may also authenticate directly to one of the other components integrated in the platform, residing on a same Web server, such as the Sympa[5] mailing list manager or the Twiki[6] Wiki manager in PicoForge version.

[3] http://shibboleth.internet2.edu/

[4] http://www.internet2.edu/

[5] http://www.sympa.org/

[6] http://www.twiki.org/

3.1 Standard authentication Scheme in phpGroupware

In general, phpGroupware handles access-permissions to its applications in an autonomous way, for locally authenticated users, basing itself on a local "account", stored in a "directory" operated for instance by a RDBMS like MySQL, or by a LDAP directory. Access to applications is a three phases process. First, the authentication verifies that the user is the owner of an account. Second, the user's profile is determined. Finally, a work session is created and access to modules is granted.

Depending on the physical implementation of the local accounts directory (MySQL, LDAP, ...), it is possible to share the user's profile with other applications deployed on the same networked environment, providing that necessary administrative policies are adopted, and that custom technical adapters are developed, or configuration decisions are taken. There's, at the moment, no “elegant” SSO service facility, that would allow phpGroupWare to grant, to users already known in other parts of the organisation's information system, a "transparent" access to its applications.

3.2 Shibboleth+Apache as an integrator and SSO service

We need SSO service with other applications deployed throughout our company, outside the platform, to which users will have already authenticated. Shibboleth can come and help solve the needs. Hopefully many Web applications we use, like Sympa, or Twiki are compatible with Shibboleth. Thus, Apache and Shibboleth will be able to act as the integrator of their authentications mechanisms (using a distributed Web service approach). Unfortunately, phpGroupWare's authentication mechanisms come short in such a situation. We need to develop a new identification and authentication adapter for the Apache + Shibboleth combination.

3.3 Mixed environment, and legacy

Our platform may be using a Shibboleth federation once adaptors have been added to all its applications. But, as described above, Shibboleth can't be the exclusive authentication mechanism used, due to the fact that we mix company and external users. We then need a way to “bypass” Shibboleth for some of our users. Another issue has to be solved when Shibboleth is used, the local *mapping*, in the applications, between the users that are recognised in Shibboleth, and the internal reference of the local account in the application. Of course, if Shibboleth is deployed prior to setting-up the platform such a mapping is trivial. But it gets worse if it is deployed on an existing environment with many legacy accounts already existing.

Having considered all the constraints above, we propose to integrate our platforms with Shibboleth using a flexible approach. In particular, we try to facilitate the progressive integration of existing deployed instances, thus diminishing the migration burden for administrators and users. As a result, the design of the new authentication system needs to support the following cases. New users, from the company directories, not yet having a developer account on the platform. These users will be able to create a new account in the platform. Legacy users with an account on the platform : when they have an account in Shibboleth, they will be able to *map* their legacy account to the new Shibboleth identity. Legacy users not registered in shibboleth are still be able to use the platform, as before, “bypassing” the Shibboleth SSO engine. New users external to GET, will still be able to apply for registration, as before.

The situation should be similar for any other authentication mechanism than Shibboleth, to which phpGroupWare would authenticate. We then tried to propose a standard *mapping* mechanism which would not be too specific to Shibboleth.

5 Conclusion

We have described a method for integrating phpGroupWare with Shibboleth to allow the use of SSO mechanisms, while supporting several authentication sources. The integration relies a lot on the use of Apache's authentication modules instead of proceeding with an internal phpGroupWare auth. mechanism. There are very few specifics of Shibboleth in this respect, most of the issues being the same if we use other types of Apache auth. mechanisms.

We introduced several options for configuration and adaptation to other environments, in order to achieve the most generic solution. This solution adapts to several situations, like a fully operational Shibboleth environment, or a deployed platform not in sync with the Shibboleth deployment.

Integration of the new modules developed in phpGroupWare has been implemented with the current 0.9.16.011 version, and avoids modifications of its architecture. We have successfully tested a prototype system on two phpGroupWare and Picolibre platforms, with the Shibboleth infrastructure of GET/INT.

Further tests are needed, but the proof of concept is done. And we have a migration path to an integration of deployed platforms in the IS, while keeping legacy accounts.

The SSO facility obtained will help design networks of collaborative platforms that will offer greater usability and flexibility, for a wider adoption both inside companies or among creative communities on the Internet.

Bibliography

[1] Cousin E., G. Ouvradou, P. Pucci and S. Tardieu, 2002, *PicoLibre a free collaborative platform to improve students skills in software engineering*, in: 2002 IEEE International Conference on Systems, Man and Cybernetics, Vol.1, IEEE, p. 564-568.

[2] Berger O., C. Bac, and B. Hamet, 2006, Integration of Libre Software Applications to Create a Collaborative Work Platform for Researchers at GET, International Journal of Information Technology and Web Engineering 1 (3), 2006.