Shibboleth authentication for non-web application

ANDREA BIANCINI, INFN - Sezione Milano Bicocca
andrea.biancini@mib.infn.it
FABIO FARINA, Consortium GARR
fabio.farina@garr.it
SIMON VOCELLA, Consortium GARR
simon.vocella@garr.it

Abstract Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Shibboleth, PAM, JAAS

Additional Key Words and Phrases: Shibboleth, SAML, Basic authentication, PAM, JAAS, Python

ACM Reference Format:
Biancini A., Farina F., Vocella S. 2012. Shibboleth authentication for non-web application. ACM Trans. Embedd. Comput. Syst. 9, 4, Article 39 (March 2010), 6 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

User authentication and user authorization are main tasks needed to guarantee security in IT environments. One important requirement of such security solutions, is that they must be as transparent as possibile. Current user applications, usually sit on different systems and environment. This requires a solution able to permit that each system is not required to create and manage its own credential system. For this reason single sign-on (SSO, as described in [Shaer 1995]) systems have been designed and implemented. These systems have also a strong impact on users who don't need to perform multiple logins on different systems, but could be automatically identified by all the application they are using.

One of the most widespread SSO software package is Shibboleth [Morgan et al. 2004], which implements authentication and authorization over network resources with a federated approach. Shibboleth, designed and implemented by Internet2, provides an effective solution for secure multi-organizational access to web resources. It implements widely used federated identity standards, principally OASIS' Security Assertion Markup Language (SAML [Cantor et al. 2005]), to provide a federated single sign-on and attribute exchange framework. Shibboleth also provides extended privacy functionality allowing the browser user and their home site to control the attributes released to each application.

Using Shibboleth-enabled access simplifies the management of identity and permissions for organizations supporting either users or applications. Shibboleth is developed in an open and participatory environment and is freely available. This open and collaborative nature has allowed Shibboleth to become the heart of many solutions for single sign-on and Authentication & Authorization Infrastructures (AAI) in several identity federations.

The architecture of Shibboleth, described in [Erdos and Cantor 2005], identifies two main components: Identity Providers (IdPs) which authenticate the users and supply user authorization attributes; Service Providers (SPs) which consume user attributes and provide access to the secure contents.
The communication excanges between the user, the IdP and the SP are all base don SAML assertions and transit over HTTP channels. This implementation architecture permits Shibboleth to works seamlessly inside user web-browsers. However, this approach is strongly web-centric, and for this reason does not adapt well to provide SSO authentication to non web-based application.

In this article a solution is presented to permit Shibboleth authentication for non web-based applications. The traditional login scheme used by Shibboleth to authenticate users is implemented inside a web-borser. The user, when trying to log to a web resources, is redirected to the IdP to perform authentication and obtain authorization assertions via his session metadata. The use case implemented within this article solves the problem of authenticatin a user and retrieving his metadata information from Shibboleth without using a web-broser. This solution could be integrated in traditional client-based application (for instance application written in Java or Python) to obtain a secure way to manager user logins.

This article will describe the architecture and the technical implementation of this authentication solution. As an example of possible uses of this work, different software implementations will be described showing how to integrate the shibboleth

AAI mechanisms inside several client applications (Linux platform, Java JAAS, Python).

The rest of this paper is organized as follows: the first section contains a quick overview of related works; the second section will present the architecture of the solution presented with this article; the third section will present some details about the technical implementation in the different software platform; the last section presents conclusions and future works.

2. RELATED WORKS

To solve this problem of AAI integration between Shibboleth and non-web application the more general and wide approach is the Moonshot project ([Howlett and Hartman 2005]). This project aims explicitly to develop a single unifying technology for extending the benefits of federated identity to a broad range of non-Web services. The Moonshot project proposes a solution to the AAI problem very complete and articulated.
It leverages different technologies such as Kerberos (a computer network authentication protocol which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner), the GSS library (Generic Security Services libraries for programs to access security services) and a Radius server (Remote Authentication Dial In User Service, a networking protocol that provides centralized Authentication, Authorization, and Accounting management).
This solution is for sure very complete and trusthworty. It implies however a quite complex architecture and requires a lot of changes on client side, SPs and IdPs. This suggests a difficult introduction of this approach into real existing federations.

The need to leverage Shibboleth identity federations in non-web application has been particularly felt in Grid Computing ([Kesselman and Foster 1998]) services. In the field of research, in fact, in the last years strong investments have been done to build a global e-Infrastructure leveraging Grid Computing technologies. This infrastructure supports the execution of scientific computation and the storage of relevant research data.
The grid, since its inception, has introduced specific solutions (inspired to the original article [Foster et al. 1998]) to security problems. The new Shibboleth SSO for web-application somehow lies next to the Grid AAI, thus the idea to integrate the two security systems is natural and potentially very beneficial.
Due to the complexity of the Moonlite project, which would add to the intrinsic complexity of grid security schemas, different solutions have been adopted in this field.
In grid environments, this problem has been approached with the goal to try and integrate the existing grid security infrastructures with Shibboleth mechanisms. These solutions, described in both articles [Wang et al. 2009] and [Jensen et al. 2007], permit the users to transparently move between Shibboleth application and Grid environments using one single set of credentials. They work with specific software middlewares able to log in the user and then manage both the Shibboleth authentication assertions and the grid certificates used to access the different environments where applications sit.
Solutions like those, are very interesting in that they really ease the user experience and provide effective results. However it must be noted that are well fit for the problem of integrating Shibboleth with grid security and can hardly be extended to other fields of application. They are not able to provide a solution for the AAI needs of non-web application not running on a grid environment.

The approaches described show opposite characteristics. On one side a very complex and articulated mechanism has been proposed, on the other side very specific and poorly adaptable solutions have been proposed to solve pragmatic problems.
In this paper a different approach will be presented. The proposed approach would try and address general problems of authentication, but will try to do it with few interventions on the requirements of SPs and IdPs.

3. ARCHITECTURE

The solution proposed leverages the standard authentication and authorization mechanisms implemented in Shibboleth. It executes a login to the IdP using HTTP Basic mechanism ([Franks et al. 1999]) over https. The authentication process of Shibboleth works as described in figure 1.

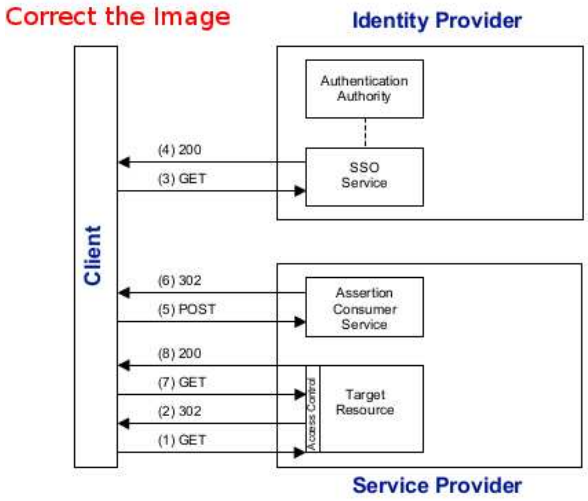

Fig. 1. Shibboleth login procedure

The solution described in this article follows the same schema, but with the following clarifications:

(1) The original request is directed to a web-page on the SP side that is behind Shibboleth authentication. This page will produce a list of rows key=value containing the metadata information to be passed to user session after login.
(2) The webserver on the SP answers with a redirect operation to a page on the IdP.
(3) The client follows the redirect and opens the IdP login page.
(4) This request obtains an answer “authorization required” asking the client to authenticate via HTTP Basic.
(5) The client authenticates performing the same request with the proper HTTP header containing username and password for the user. The answer to this request contains a Set-Cookie instruction with which the IdP creates a cookie on the client containing a couple of keys permitting the client to reconnect with the Shibboleth session created.
(6) The IdP communicates with the SP to create a valid session for the user with all the metadata of the user session.

(7) The client opens the original URL providing the cookie set by the IdP after the authentication. This requeste gets the page with the key=value rows used to initialize the user session on the client side.

This mechanism simulates what usually happens in the browser of a user authentication to a Shibboleth web-application. All this interactions are, however, hidden to the user who is not aware of the underlying HTTP dialogue to permit the authentication.

4. IMPLEMENTATIONS

4.1. PAM and NSS modules

4.2. JAAS module

4.3. Python

5. CONCLUSION AND FUTURE WORKS

REFERENCES

CANTOR, S., KEMP, J., PHILPOTT, R., AND MALER, E. 2005. Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0. Tech. rep. 03.

ERDOS, M. AND CANTOR, S. 2005. The Shibboleth architecture. `http://shibboleth.internet2.edu/`.

FOSTER, I., KESSELMAN, C., TSUDIK, G., AND TUECKE, S. 1998. A security architecture for computational grids. In *Proceedings of the 5th ACM conference on Computer and communications security*. CCS '98. ACM, 83–92.

FRANKS, J., HALLAM-BAKER, P., HOSTETLER, J., LAWRENCE, S., LEACH, P., LUOTONEN, A., AND STEWART, L. 1999. Http authentication: Basic and digest access authentication. `RFC 2617`.

HOWLETT, J. AND HARTMAN, S. 2005. Project moonshot. Tech. rep. 07.

JENSEN, J., WALLOM, D., SPENCE, D., TANG, K., MEREDITH, D., AND TRETHEFEN, A. 2007. Shibgrid, a shibboleth based access method for the national grid service. In *UK e-Science All Hands Meeting*.

KESSELMAN, C. AND FOSTER, I. 1998. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers.

MORGAN, R. L., CANTOR, S., CARMODY, S., HOEHN, W., AND KLINGENSTEIN, K. 2004. Federated security: The shibboleth approach. *EDUCAUSE Quarterly 27,* 4, 12–17.

SHAER, C. 1995. Single sign-on. *Network Security 1995,* 8, 11–15.

WANG, X. D., JONES, M., JENSEN, J., RICHARDS, A., WALLOM, D., MA, T., FRANK, R., SPENCE, D., YOUNG, S., DEVEREUX, C., AND GEDDES, N. 2009. Shibboleth access for resources on the national grid service (sarongs). In *Proceedings of the 2009 Fifth International Conference on Information Assurance and Security - Volume 02*. IAS '09. IEEE Computer Society, 338–341.

Received September 2012; revised September 2012; accepted September 2012