New single sign-on mechanism for EFDA Integrated Tokamak Modelling Task Force portal

V.F. Pais and ITM-TF contributors*

National Institute for Laser Plasma and Radiation Physics, Lasers Dept., Magurele, Romania, Association EURATOM/MEdC

The concept of single sign-on allows users to logon only once and use a token to access different service providers in distributed computer networks. This is what happens when a user is accessing a portal that comprises several applications.

The EFDA Integrated Tokamak Modeling Task Force (ITM-TF) was set up in 2004 with the long term aim to provide the EU with a suite of codes necessary for preparing and analyzing future ITER discharges, with the highest degree of flexibility, confidence and reliability.

One of the resources available to users of ITM-TF is the ITM Portal. It is comprised of several web applications related to fusion research and acts as an interface to other task force resources, especially the ITM Gateway [1] cluster.

At the beginning of the ITM Portal, Shibboleth was used as the single sign-on mechanism [2]. However, after several years it has become apparent that a new system was needed in order to offer more flexibility for integrating applications in the portal. Even though Shibboleth performed very well its single sign-on functions, some functionality was missing from the overall system, such as a central administrative interface. This can be easily explained by the fact that Shibboleth was designed to operate in a federated environment where various applications are placed on servers with different administrative policies. Therefore, Shibboleth is a good choice for such systems, making it even inter-operable with other federation technologies, but it was a bit harder to administrate given a local installation, requiring only to link applications under the same administrative policies.

Another issue with Shibboleth was that at the time it was installed it had no support for global logout. This meant that a user that used multiple web applications could press the logout button inside one application, but still remained logged in for other applications. Thus an attack would have been possible given access to the user's browser. Another person could simply switch to the application where the user was still logged on and work on the user's behalf.

* http://www.efda-itm.eu/

Furthermore, the Shibboleth system had a requirement to install an Apache module in order to handle authorization requests. Even though this is not necessarily an issue, it made the administrative task a difficult, especially with regard to system updates. Every update involving the web server must first be tested for compatibility with Shibboleth.

When the decision to drop Shibboleth was taken, other single sign-on mechanisms were considered, such as PAPI [3]. However, most of the mechanisms studied are best suited for federations. Therefore, it was decided to implement a lightweight single sign-on system especially for the ITM Portal.

In addition to offering solutions to the already identified problems, the new system had to include the possibility of using multiple authentication sources. This is related to the fact that previously only users with an ITM Gateway account were able to access any ITM-TF resources. However, in recent years some accounts were needed for external users with limited access to tools such as subversion. Since these users do not posses a machine account, there must be a separate authentication and authorization system available, integrated with web based applications.

Because the system needed to be as simple as possible, without any additional requirements on the operating system, it was decided to implement it purely in PHP. The internal structure is similar to that of any single sign-on mechanism: one identity provider and several service providers.

The identity provider (IdP) is in charge of authenticating the client and retrieving its access rights. Due to the requirement of having multiple user databases, a priority list was used to make sure machine accounts are given priority over external user accounts. The IdP is the only component actually accessing the user databases. Any other components are in direct communication with the IdP whenever authentication or authorization is required.

The service provider (SP) is installed locally on each server providing access to web applications. It is a PHP script with a configuration file, also in PHP format, that must be placed either in the root directory of the web application or in the root directory of the web folder. This way there is no file required outside of the web directory and no modification to be made to the server's configuration. Furthermore, there is no dependency on a certain PHP version, since only the most basic features are being used, thus making it compatible with older versions.

When a user tries to access a certain resource, access is intercepted by the SP script and the configuration file is checked. If the resource being accessed is deemed to be restricted, the user is redirected to the IdP for authentication. After a successful logon, the user is sent back to the SP and access rights are received from the IdP using a back-channel. This means that sensitive data never gets to the user's browser, since a direct communication channel is used between SP and IdP. The SP now sets a session cookie and caches the data received from the IdP in a special directory on the server. Once the user tries to access another protected resource on the same SP, it already knows who the user is and if it has the right to access it or not.

For applications aware of single sign on technologies, the SP sets the REMOTE_USER environment variable. This is a standard way of letting the applications know who is accessing them. This way it was possible to migrate existing applications to the new system with no modifications, since the same variable was being used with Shibboleth. Even more, the SP sets another variable to the groups the user belongs to, thus making this information available to applications, and allowing to easily develop more tight integrations without actually using any of the single sign on provided functionality.

Using the fact that all the ITM-TF applications are in the same administrative domain, only on different servers, and are quite limited in number, it was possible to implement a central logout mechanism. This can be called from any application by redirecting the user to a logout page on the IdP. This in turn will redirect the user to all the service providers each one being allowed to completely cleanup all the session data. Of course, this implementation would not have been possible in a federated environment with a large number of service providers. However, given the size of the ITM environment this works very well and completely removes any potential security related problems introduced by session cookies remaining in the user browser.

With other proposed single logout mechanisms, usually the back channel is used for expiring sessions locally on the SP, without informing the user browser. Thus, the user still retains his session cookie, but in theory it can't be used on the SP, since it is no longer valid. However this is not always the case, since the back channel communication is not visible to the user and thus he can not be sure when then session is actually expired, in case something goes wrong on one of the service providers.

With the ITM implementation, once the user sees the message indicating he is logged out, all the information regarding his access rights, from the point of view of the single sign on system, was purged from all service providers and from his own browser.

The implemented system is a general purpose single sign on mechanism. It is well suited especially for small to medium size web environments, such as those present in a single organization. Nevertheless, all the functionality can be used in a federation, including the single logout feature, as long as the number of service providers is kept under control.

The fact that the logout feature completely destroys the session data on the user computer is particularly useful in secure environments. Furthermore, by integrating multiple authentication sources, the system is well suited to be used in environments with a mix of internal and external users. Even more, being written completely in PHP with no external dependencies, the system is independent of the operating system used or its update level, as long as it provides support for PHP and Apache web server.

Acknowledgments

This work, supported by the European Communities under the contract of Association between EURATOM- MEdC was carried out within the framework of the Task Force on Integrated Tokamak Modelling of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] "Gateway: new High Performance Computing facility for EFDA Task Force on Integrated Tokamak Modelling", F. Iannone, B. Guillerminet, F. Imbeaux, G. Manduchi, A. Maslennikov, V. Pais and P. Strand, 2010, Fusion Engineering and Design 85 (2010), pp. 410-414

[2] "Enabling remote access to projects in a large collaborative environment", V.F. Pais, S. Balme, H.S.Akpangny, F. Iannone, P. Strand, 2010, Fusion Engineering and Design 85 (2010), pp. 633-636

[3] “PAPI based federation as a test-bed for a common security infrastructure in EFDA sites“, R. Castro, J. Vega, A. Portas, D.R. López, S. Balme, J.M. Theis, P. Lebourg, H. Fernandes, A. Neto, A. Duarte, F. Oliveira, F. Reis, K. Purahoo, K. Thomsen, W. Schiller, J. Kadlecsik, Fusion Engineering and Design, Volume 83, Issues 2–3, April 2008, Pages 486-490

[4] “Ensuring information assurance in federated identity management”, Dongwan Shin, IEEE International Conference on Performance, Computing, and Communications, 2004, Pages 821-826