Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 5, Issue. 1, January 2016, pg.53 – 56

Different Framework for Single Sign On (SSO)

Kurhe Bhagwan Subhash[1], Prof. Kahate S.A.[2]

[1]Sharadchandra Pawar College Of Engineering, Otur, Tal-Junnar Dist-Pune
b.kurhe@gmail.com
[2]Sharadchandra Pawar College Of Engineering, Otur, Tal-Junnar Dist-Pune
Sandip.kahate@gmail.com

Abstract: Single sign-on (SSO) is a method of access control that enables a user to log in once and gain access to the resources of multiple software systems without being prompted to log in again. Single sign-off is the reverse process whereby a single action of signing out terminates access to multiple software systems. As different applications and resources support different authentication mechanisms, single sign-on has to internally translate to and store different credentials compared to what is used for initial authentication. This paper gives clear idea about different framework used for SSO
Keywords: Single Sign On(SSO)

1. Introduction

Historically a distributed system has been assembled from components that act as independent security domains. These components comprise individual platforms with associated operating system and applications.

These components act as independent domains in the sense that an end-user has to identify and authenticate himself independently to each of the domains with which he wishes to interact. This scenario is illustrated above. The end user interacts initially with a Primary Domain to establish a session with that primary domain. This is termed the Primary Domain Sign-On in the above diagram and requires the end user to supply a set of user credentials applicable to the primary domain, for example a username and password. The primary domain session is typically represented by an operating system session shell executed on the end user's workstation within an environment representative of the end user (e.g., process attributes, environment variables and home directory). From this primary domain session shell the user is able to invoke the services of the other domains, such as platforms or applications. To invoke the services of a secondary domain an end user is required to perform a Secondary Domain Sign-on. This requires the end user to supply a further set of user credentials applicable to that secondary domain. An end user has to conduct a separate sign-on dialogue with each secondary domain that the end user requires to use. The secondary domain session is typically represented by an operating system shell or an application shell, again within an environment representative of the end user. From the management perspective the legacy approach requires independent management of each domain and the use of multiple user account management interfaces. Considerations of both usability and security give rise to a need to co-ordinate and where possible integrate user sign-on functions and user account management functions for the multitude of different domains now found within an enterprise. A service that provides such co-ordination and integration can provide real cost benefits to an enterprise through: reduction in the time taken by users in sign-on operations to individual domains, including reducing the possibility of such sign-on operations failing Improved security through the reduced need for a user to handle and remember multiple sets of authentication information. Reduction in the time taken, and improved response, by system administrators in adding and removing users to the system or modifying their access rights.

Improved security through the enhanced ability of system administrators to maintain the integrity of user account configuration including the ability to inhibit or remove an individual user's access to all system resources in a co-ordinated and consistent manner.

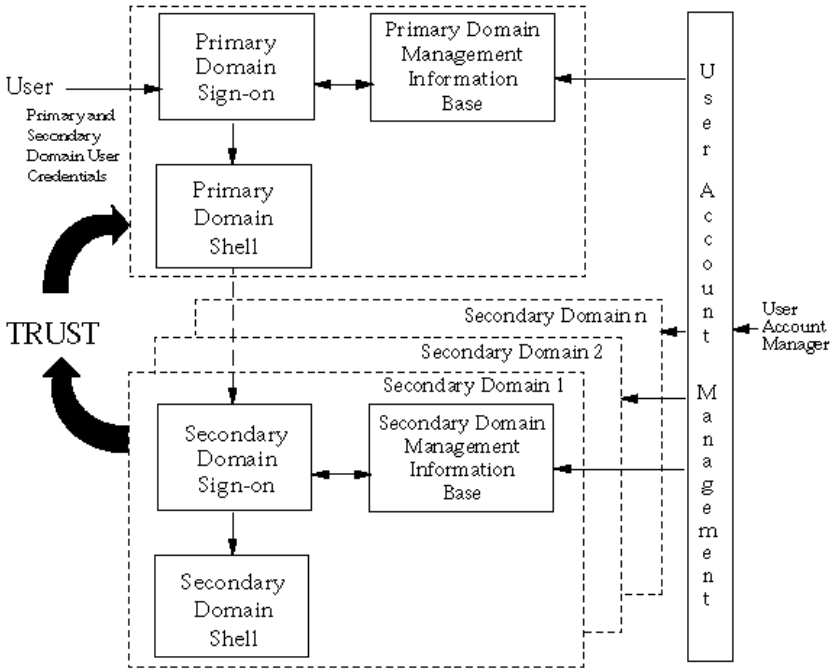

Fig.1.1 **Single User Sign-On To Multiple Services**

Such a service has been termed Single Sign-On after the end-user perception of the impact of this service. However, both the end-user and management aspects of the service are equally important. This approach is illustrated in the diagram above. In the single sign-on approach the system is required to collect from the user as, part of the primary sign-on, all the identification and user credential information necessary to support the authentication of the user to each of the secondary domains that the user may potentially require to interact with. The information supplied by the user is then used by Single Sign-On Services within the primary domain to support the authentication of the end user to each of the secondary domains with which the user actually requests to interact.

The information supplied by the end-user as part of the Primary Domain Sign-On procedure may be used in support of secondary domain sign-on in several ways:

- Directly, the information supplied by the user is passed to a secondary domain as part of a secondary sign-on.
- Indirectly, the information supplied by the user is used to retrieve other user identification and user credential information stored within the a single sign-on management information base. The retrieved information is then used as the basis for a secondary domain sign-on operation.
- Immediately, to establish a session with a secondary domain as part of the initial session establishment. This implies that application clients are automatically invoked and communications established at the time of the primary sign-on operation.
- Temporarily stored or cached and used at the time a request for the secondary domain services is made by the end-user.

From a management perspective the single sign-on model provides a single user account management interface through which all the component domains may be managed in a coordinated and synchronized manner.

Significant security aspects of the Single Sign-On model are:

- The secondary domains have to trust the primary domain to:
- The authentication credentials have to be protected when transferred between the primary and secondary domains against threats arising from interception or eavesdropping leading to possible masquerade attacks

2. Open SSO

The Open Web SSO project (OpenSSO) provides core identity services to simplify the implementation of transparent single sign-on (SSO) as a security component in a network infrastructure. OpenSSO provides the foundation for integrating diverse web applications that might typically operate against a disparate set of identity repositories and are hosted on a variety of platforms such as web and application servers. This project is based on the code base of Sun Java™ System Access Manager, a core identity infrastructure product offered by Sun Microsystems

3. Active Directory Federation Services (ADFS)

Active Directory Federation Services (ADFS) is a standards-based service and a component in Microsoft Server family of operating systems, starting from Windows Server 2003 R2 with ADFS 1.0. The latest available version is ADFS 3.0 on Windows Server 2012 R2. ADFS provides web Single Sign-On within the boundaries of 8 organisational Active Directory forest infrastructure. ADFS also provides identity federation between trusting organisations through inter-forest trust relationships [8], [9]. The SAML 2.0 standard is supported by ADFS, as well as OAuth 2.0 in the most recent version, ADFS 3.0 [10], [11]. Software developers can make use of provided libraries, such as Windows Azure Active Directory Authentication Library

(ADAL), to enable their client applications to use the provided SSO capabilities to access one or more web-based APIs, provided by Resource Servers [12]. The Microsoft Developer Network page includes GitHub links for native ADAL libraries for different mobile or static platforms, including iOS, OSX and Android [12].

4. Single Sign-On categories and frameworks

When it comes down to different protocols and frameworks currently present in the industry, one can only think of a few names. These are Security Assertion Markup Language (SAML), OAuth, OpenID and very recently, OpenID Connect. Here, we are going to briefly describe these different standards. Further detail on the chosen ones will be included in section 5. It needs to be particularly mentioned that, one of the major uses of the aforementioned frameworks and protocols is the handling of the authorization requirements of Application Programming Interfaces (APIs). This is especially true about web applications. Whether it is a Google, or Facebook type of on-line service provided as an API, or it is a company's internal web application API, there will be a need to provide delegated access to certain resources using a API as an interface. Thus, the accessing application has to be authorized for delegated access. In legacy applications, the authorization was done by giving the user-name and password of the owner to the application [5]. One can easily imagine how tedious it will be to work with a number of applications accessing different APIs and asking the credentials required for each authorization separately. It also creates higher security risks, since the owner has no choice but to trust these applications [5].

5. SAML 2.0

SAML was created by OASIS Security Services Technical Committee and its current version is SAML 2.0, dating back to 2005 [18]. SAML relies on XML-based data to transfer authentication and authorization details. Using SAML, users, applications and services can exchange identity information [2]. This is done through SAML Assertions, which are compressed, encoded and possibly encrypted XML nodes [1].

6. OAuth 2.0

The latest iteration of OAuth, formalized in 2012, is the version 2.0 [19]. As one can imagine, it is much more accommodating to current trends and needs in the industry, as we will see in section 5. OAuth includes the notion of Access Token as the mechanism of choice for allowing access to restricted resources. In other words, an Access Token is the authorization issued to a client [19].

7. OpenID 2.0

At the time of this writing, the latest iteration of OpenID from 2007, the version 2.0, has become obsolete by the introduction of OpenID Connect. We will briefly mention a general description here and move on to OpenID Connect. Both standards are created by OpenID Foundation. OpenID focuses on providing decentralized authentication for end-users, throughout cooperating websites. Meaning, users can use a single identity to authenticate 10 against different websites. The idea is called Bring Your Own Identity (BYOI) and it is in wide use today. Standard HTTP(S) requests are used for this purpose [12].

8. OpenID Connect 1.0

OpenID Connect 1.0, finalized on February 2014, adds an identity layer to OAuth 2.0, enabling the verification of an end-user. This is done by using the data from an authentication, which an involved OAuth 2.0 Authorization Server performs [1]. Considering the similarities between OAuth 2.0 and OpenID Connect 1.0 and given the fact that the current implementation at Eurostep AB is using OAuth 2.0, it would be enough for our purposes to focus on OAuth 2.0.

9. The importance of BYOD

Bring Your Own Device (BYOD) and more generally, Bring Your Own Technology (BYOT), are descriptive terms for connecting employee owned devices or technologies (hardware or software) to company infrastructure. The practice is considered a liberty for employees, but actually involves far reaching benefits and risks for both parties. For instance, having employee owned devices connecting to business services annuls the requirement of buying company owned hardware, as well as certain amount of time required for the relevant training. On the other hand, a refusal by the employer would create the dissatisfaction of employees [13]. According to statistics mentioned by Miller et al., there was a 35 percent increase in smart-phone ownership rate between May 2011 and February 2012 in United States. Furthermore, 71 percent of people between the ages of 25 and 34 own smart-phones [13]. Since having two or more devices is far from being convenient or practical, the BYOD approach sounds like an eventuality. On the other hand, the risk to security (with regards to company data) and the risk to privacy (with regards to employee personal data) goes hand in hand with the adoption of BYOD [13]. As a result, a feasible solution can be the use of personal hardware as a portal to connect to a centralized data storage or service. Cloud computing can be an example. This is where Single Sign-On comes in and could provide us with the means for the necessary access control.

CONCLUSION

This paper study different frameworks of the SSO in details.

References

[1] Z. Dennis. (May 9, 2013). Choosing an SSO strategy: SAML vs OAuth2, Mutually Human, [Online]. Available: http://www.mutuallyhuman.com/blog/2013/05/09/choosing- an- sso- strategy- saml- vs- oauth2/ .

[2] N. Ranjbar and M. Abdinejadi, "Authentication and authorization for mobile devices", Aug. 6, 2012. [Online]. Available: https://gupea.ub.gu.se/handle/2077/30043

[3] D. Todorov, Mechanics of User Identification and Authentication. Boston, MA,USA: Auerbach Publications, 2007, isbn: 1420052195.

[4] C. P. Pfleeger and S. L. Pfleeger, Security in computing, 4th. Prentice Hall Professional, 2007, isbn: 0132390779.

[5] R. Boyd, Getting started with OAuth 2.0. O'Reilly Media, Inc., 2012, isbn: 9781449311605.

[6] J. Catone. (2008). Bad form: 61% use same password for everything, Read Write, [Online]. Available: http://readwrite.com/2008/01/17/majority_use_same_password

[7] (2010). Introduction to single sign-on, THE Open GROUP, [Online]. Availablehttp://www.opengroup.org/security/sso/sso_intro.htm.

[8] (Aug. 22, 2005). Introduction to ADFS, TechNet, Microsoft, [Online]. Available: http://technet.microsoft.com/en-us/library/cc786469(v=ws.10).aspx

[9] (2008). Active directory federation services, Developer Network, Microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/library/bb897402.aspx

[10] A. Simons. (Apr. 22, 2013). Developer preview of OAuth code grant and AAL for windows store apps, Active Directory Team Blog, [Online]. Available: http://blogs.technet.com/b/ad/archive/2013/04/22/developer-previewof-oauth-code-grant-and-aal-for-windows-store-apps.aspx.

[11] U. Hegde. (Jul. 10, 2013). Extending device support in active directory, Active Directory Team Blog, [Online]. Available: http://blogs.technet.com /b / ad / archive / 2013 / 07 / 10 / extending - device - support - in - active-directory.aspx .

[12] (Apr. 9, 2014). Developing modern applications using oauth and active directory federation services, Developer Network, Microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/library/dn633593.aspx .

[13] K. Miller, J. Voas, and G. Hurlburt, "Byod: security and privacy considerations",IT Professional, vol. 14, no. 5, pp. 53–55, Sep. 2012, issn: 1520-9202. doi: 10.1109/MITP.2012.93. 30

[14] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple object access protocol (soap) 1.1", The World Wide Web Consortium (W3C), 2000. [Online]. Available: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ .

[15] "Bindings for the OASIS security assertion markup language (SAML) v2.0",OASIS, 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf.

[16] C. Pautasso and E. Wilde, "Restful web services: principles, patterns, emerging technologies", in Proceedings of the 19th International Conference on World Wide Web, ser. WWW '10, Raleigh, North Carolina, USA: ACM, 2010, pp. 1359–1360, isbn: 978-1-60558-799-8. doi: 10 . 1145 / 1772690 . 1772929. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772929.

[17] D. Syer. (Oct. 9, 2012). Securing RESTful web services with OAuth2, Cloud Foundry Blog, [Online]. Available: http://blog.cloudfoundry.org/2012/10 / 09 / securing - restful - web - services - with - oauth2/ .

[18] "Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0", OASIS, 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[19] D. Hardt, "The OAuth 2.0 authorization framework", Internet Engineering Task Force (IETF), 2012. [Online]. Available: https : / / tools . ietf.org /html/rfc6749

[20] "Openid authentication 2.0 - final", OpenID Foundation, 2007. [Online]. Available:http : / / openid . net / specs / openid - authentication - 2 _ 0 . html

[21] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, "Openidconnect core 1.0", OpenID Foundation, 2014. [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html

