International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

SSO (Single Sign On) Implementation

Parul Garg[1], Dr. Yashpal Singh[2]

[1]Research Scholar Department of Computer Science & Engineering, Ganga Institute of Technology and Management, Kabalana, Jhajjar, Haryana

[2]Associate Professor, Department of Computer Science & Engineering, Ganga Institute of Technology and Management, Kabalana, Jhajjar, Haryana

Abstract: ***This paper is a case study on a project to provide a Single Sign On (SSO) solution to web based applications that use the mainframe as the data store. This is a process that allows an authenticated user to enter their credentials to access various application. Conversely the multiple access can be terminated by the action of single sign-out the system. The paper begins with a thorough description of the high level business requirements.***

Keywords: SSO, DNS, LDAP, Password synchronization, Web SSO, Federation

1. Introduction

Single sign-on (**SSO**) is a process that allows an authenticated user to enter their credentials to access various application. Conversely the multiple access can be terminated by the action of single sign-out the system. This process allows the user to access only those applications for which they terminates further prompts during a specific session when they switch the application. The process is done using Lightweight Directory Access Protocol (**LDAP**) and stores **LDAP** databases on servers also called directory servers.[1] The same process can be achieved using IP protocol with cookies only when the sites share a common DNS domain. [2] Refer fig:1.

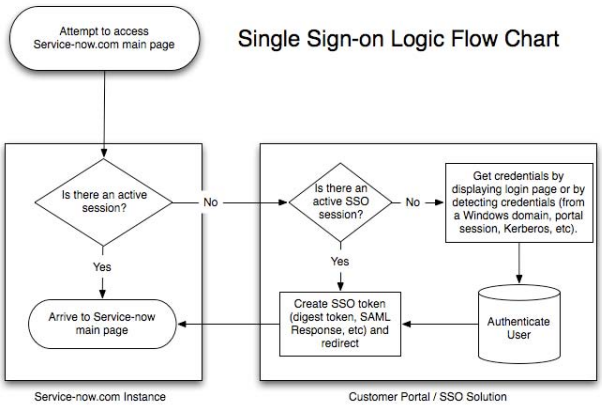

Fig: 1 Single Sign-on Logic Flow Chart

Usually the system requires the credentials for each and every application, but by using SSO one can simplify this process by providing same credentials from the directory server to the configured applications by passing them an authenticated token. Other different authentication schemes like OAuth, OpenID, OpenID Connect and Facebook Connect, [6] [7] which require the user to enter their login credentials each time they access a different site or application need not to be confused with SSO as they involve a totally different schemes. Refer fig:2

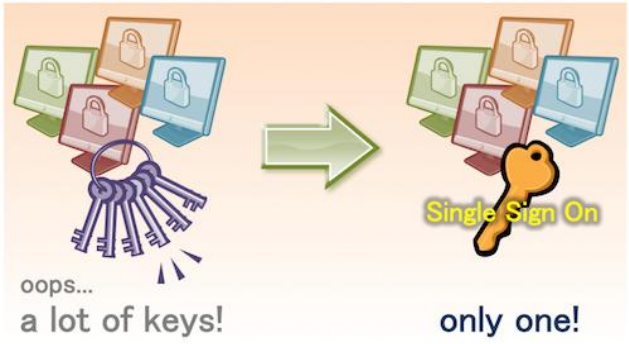

Fig: 2 Working of SSO

2. History of SSO

S.No	Year	Work Undergone Till Present
1.	2016	Single Sign-On First
2.	2016	Censorship, external authentication, and other social media lessons from China's Great Firewall
3.	2016	Signing Me onto Your Accounts through Facebook and Google: a Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services
4.	2015	The Censorship Effect
5.	2014	SSO and LDAP Authentication
6.	2014	OpenID versus Single-Sign-On Server
7.	2013	Single Sign On Authentication

3. Advantages of SSO

- Users tend to choose stronger **password**s, and hence the need for multiple **password**s and change **synchronization** is eliminated.
- At the point of user entry, Inactivity timeout and attempt thresholds are applied.
- It disables all the user accounts who have terminated and hence improves the effectiveness of the system.
- This enhances an administrator's ability to manage multiple users and their related configurations to all the associated systems.
- Users need to remember only a single set of the credentials i.e. username and password.
- This process thus improves the security and the credentials are preserved with the user as they need not write them

down anywhere as remembering single credential is easy thus simplifying the user logon process.
- It reduces the time taken by users to log onto different applications and platforms easily.

4. SSO Categories

Main categories are:

(i) **Password synchronization**
Accessing multiple systems require unique usernames but a common **password**.

(ii) Enterprise **SSO**
Enterprise **SSO** allows every individual application to simplify its password complexity rules but sometimes it is difficult to implement this approach as it leads to integration difficulties due to small pockets of non-coverage. [15]

(iii)**Web SSO** Federation
To create the relationship between different organizations, **Web SSO** Federation is used, that further uses standards such as SAML [16] or Windows Federated Authentication.

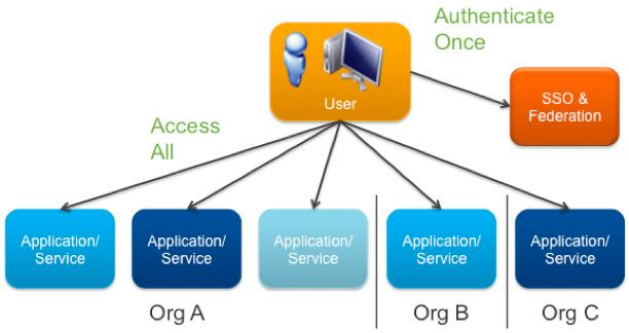

Figure 3: Approaches used with SSO

The person who is logging in from an organization is asked a question in return for the approval, so it is **SSO** because usually the token can be re-used across numerous federated applications. Refer fig:3

5. Implementing SSO

5.1 Recommendations for companies planning an SSO implementation

- Prioritize which systems group of users are important and then choose the right **SSO** approach for that\those particular user or system.
- Analyse particulars that exist in the company such as Active Directory, as much as possible.

If the company implements Federation, consider the thing as a whole to select an **SSO** scheme that doesn't restrict the implementation to a small pocket of systems or group of users. Refer fig:4

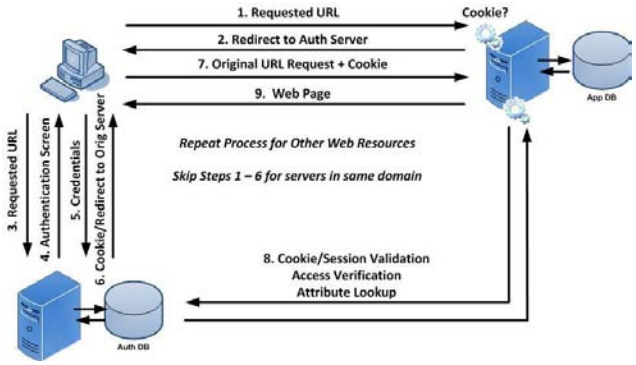

Figure 4: Implementation to a small pocket of systems

6. Benefits

6.1 Benefits of using single sign-on include:

- Use of single username and p**assword** eliminate use of multiple credentials.
- Saving lot of time in entering multiple passwords since one needs to enter single username and password.
- Reducing expense due to lesser number of IT help desk calls related to **password**s. [3]

SSO has a centralized authentication servers system to which all other applications and systems are connected for authentication purposes and combines this with different techniques to ensure that users do not have to actively enter their credentials more than once.

7. How SSO deals with Problem Arises

When web development teams face one problem, example: you have developed an application at domain X and now you want your new deployment at domain Y to use the same login information as the other domain. In fact, you want more: you want users who already logged-in at domain X to be already logged-in at domain Y. [4][5] This is what SSO is all about. Refer Fig 5

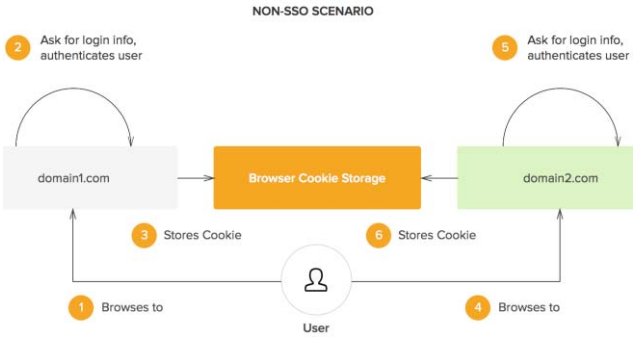

Figure 5: Non SOS Scenario

The solution to this problem is to share session information across different domains. However, for certain security reasons, browsers applies a policy which the same origin policy. This policy dictates that cookies can only be accessed by its creator (i.e. the domain that originally requested the data to be stored). In other words, domain X cannot access cookies from domain Y or vice versa. This is what SSO shares session information across different domains. Refer fig 6

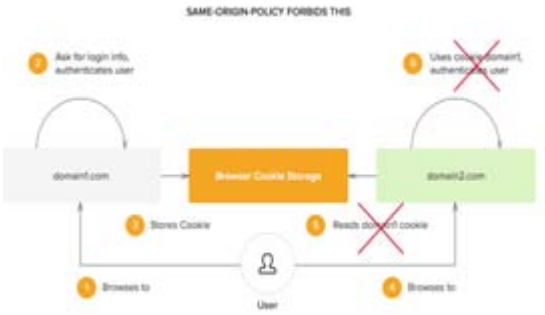

Figure 6: Same Origin Policy Forbids

Different SSO protocols share session information in different ways, but the base is the same: there is a central domain, through which authentication is performed, and then the session is shared with other domains in some way. For example, the central domain generates a signed JSON Web Token (which may be encrypted using JWE). This token is further passed to the client and also to different other domains. The token can be passed to the original domain by a redirect and contains all the information needed to identify the user for the domain requiring authentication. When the token is signed, it cannot be further modified by the client in any other form.

8. Security

- Single point of failure[8]
- Single high-value target (attracts more attackers) [9]
- Necessary information disclosure between trusting site and **SSO** authority[10]
- Side channel attack against authentication step (theoretically; implementation dependent) [11]
- Lack of control over your user list[12]
- Yet another interface to maintain (added complexity) [13]
- You may never know how secure your system is or if there is a breach[14]
- Added cost

9. Conclusion

Decentralized systems are so common and authentication is an important aspect of them. Using SSO a big problem is solved: Managing large population across the globe of applications and services. When implementing authentication for a new application, consider integration it with the SSO. The results of the analysis conducted for this paper indicate that technology issues are the main factors implementing **SSO** and MFA within organisations; this is partly due to complexities of existing technical infrastructures and workflow processes. SSO is here to stay.

10. Related Work

- Client side implementation with plug-ins for various services and protocols.
- Claim based system and application federations
- Identity and access management solution
- Identify infrastructure for development
- Protocol and SSO client/server architecture

References

[1] "SSO and LDAP Authentication". Authenticationworld.com. Retrieved 2014-05-23
[2] "OpenID versus Single-Sign-On Server" alleged.org.uk. 2007-08-13. Retrieved 2014-05-23.
[3] "Benefits of SSO". University of Geulph. Retrieved 2014-05-23.
[4] "Single Sign On Authentication". Authenticationworld.com. Retrieved 2013-05-28.
[5] "Sun GlassFish Enterprise Server v2.1.1 High Availability Administration Guide". Oracle.com. Retrieved 2013-05-28.
[6] Laurenson, Lydia (3 May 2014). "The Censorship Effect". TechCrunch. Retrieved 27 February 2015.
[7] Chester, Ken (12 August 2013). "Censorship, external authentication, and other social media lessons from China's Great Firewall". *Tech in Asia*. Retrieved 9 March 2016..
[8] *Rui Wang, Shuo Chen, and XiaoFeng Wang.* "Signing Me onto Your Accounts through Facebook and Google: a Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services".
[9] "OpenID: Vulnerability report, Data confusion" - OpenID Foundation, March 14, 2012
[10] "Facebook, Google Users Threatened by New Security Flaw". *Tom's Guide. 2 May 2014*. Retrieved 11 November 2014.
[11] "Covert Redirect Vulnerability Related to OAuth 2.0 and OpenID". *Tetraph. 1 May 2014*. Retrieved 10 November 2014.
[12] "Math student detects OAuth, OpenID security vulnerability". *Tech Xplore. 3 May 2014*. Retrieved 10 November 2014.
[13] "Facebook, Google Users Threatened by New Security Flaw". *Yahoo. 2 May 2014*. Retrieved 10 November 2014.
[14] "Covert Redirect Flaw in OAuth is Not the Next Heartbleed". *Symantec. 3 May 2014*. Retrieved 10 November 2014.
[15] "Single Sign-On First". *Evolveum. 3 July 2012*. Retrieved 2 January 2016.
[16] "MicroStrategy's office of the future includes mobile identity and cybersecurity". *Washington Post. 2014-04-14*. Retrieved 2014-03-30.