A Review on Single Sign on as an Authentication Technique

Vijay Koundinya, Shwetha Baliga

Abstract— In today's digital era, where security is extremely big concern and login are required for each small snippet of data one is trying to get or access. This has led to the industry developing SSO (Single-Sign-On). This paper presents review on SSO enabling technologies and discusses SSO architectures, and protocols associated with growing use of SSO. Finally, a time analysis is done to show the amount of time that can be reduced using SSO as an authentication technique.

Index Terms— *User Authentication, Service Authentication, Single Sign On, Single Sign On (SSO), Authentication, Multi-factor Authentication (MFA), SAML, OpenID*.

1. Introduction

Generally, Application Service Provider(ASP) provides a standard interface to a countless number of users and also a standard connection point to various application providers. With security and privacy concerns being very high in today's online climate, every application tends to use it's own authentication mechanism. A user has multiple authentication steps to access the application. In order to address the issue of user convenience while maintaining a high level of security, SSO(Single Sign-on) was developed. SSO asks the user to log in once with a which will provide the user with access to the entire set of application services without having to be authenticated over again. SSO allows for the integration of user information and also the security policy [1].

Prior to SSO, every user had to remember for each application, as each user had to perform multiple logins while accessing many applications provided by a single application service provider. This led to users sticking to simpler passwords or repeating passwords across applications, which is a security risk. SSO, in theory, allows for one to pick a more complicated password as one needs to remember only one.

The structure of this paper is outlined as follows; in Section2, this paper discusses the architecture and variants of SSO. In Section 3, discussion about SSO enabling technologies and protocols is done. Section 4 is a time analysis study of user behavior to show how SSO could improve efficiency and Section 5 concludes the paper.

2. SSO

2.1 Overview

Single Sign-on (SSO) occurs when a user logs in to one application and is then signed in to other applications automatically, regardless of the platform, technology, or domain the user is using. The user signs in only one time, hence the name of the feature (Single Sign-on).

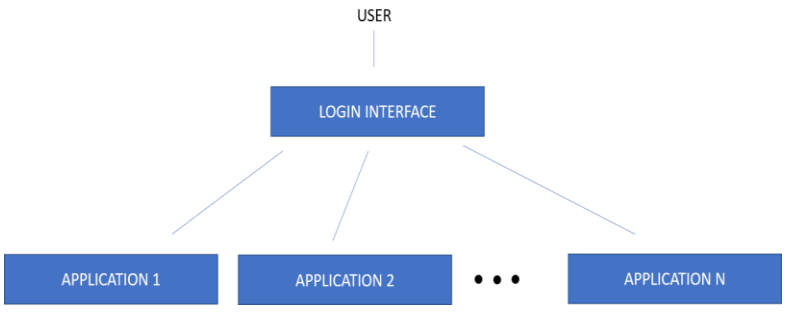

Fig. 1. gives us a pictorial description of the overview of sso as a service.

2.2 System Architecture

There are different types of SSO architectures, with different properties and infrastructures namely Secure Client-Side Credential Caching, Secure Server-Side Credential Caching, SSO with Single Set of Credentials, Public Key Infrastructure based SSO, Token based SSO. Secure Client-Side Credential Caching and Secure Server-Side Credential Caching come under SSO with multiple sets of credentials while Public Key Infrastructure based SSO and Token-based SSO come under SSO with a single set of credentials. Depending on their properties and usage, these architectures can be applied to various situations accordingly. The detailed description of these architectures is discussed below [2].

1 Secure Client-side caching: Client-side password caching allows users who are not members of the Webspace Server's domain to log on without having to enter their user name and password every time they connect to the server. With this option enabled, the Login dialog box will display a "Remember me on this Computer" checkbox. If a user selects this checkbox on the first login from the client, the next time that user logs in from that same computer, the Logon dialog box will show the User Name and Password dialog box pre-populated with the previous login. All the user needs to do to continue is click Sign In. This is an example of client-side credential caching. But, this is not an advisable practice.

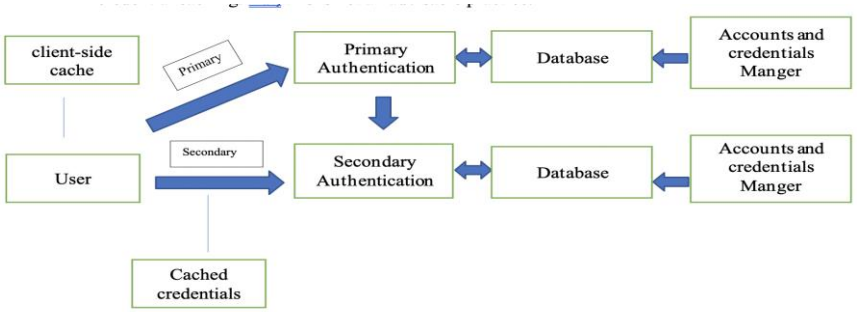

Fig 2. Server Client Side Caching

2. Server side caching - It is the same as client side credential caching, with the only difference being the credentials are cached in the sever. A central server is used to administer all the credentials and provide any necessary information directly to the application asking for it.

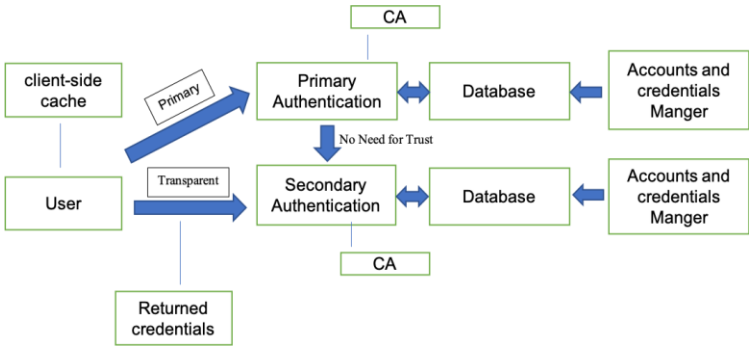

Fig 3: Server Side Caching

PKI-based Single Sign-On: In this type of SSO users and servers authenticate each other using a key pair. Both the server and the user can challenge each other by asking them to decipher messages. Therefore a mutual authentication can take place, server to user and vice versa also. The certificate authority may also defer between the user and the server as long as there exists an established trust between the two authorities

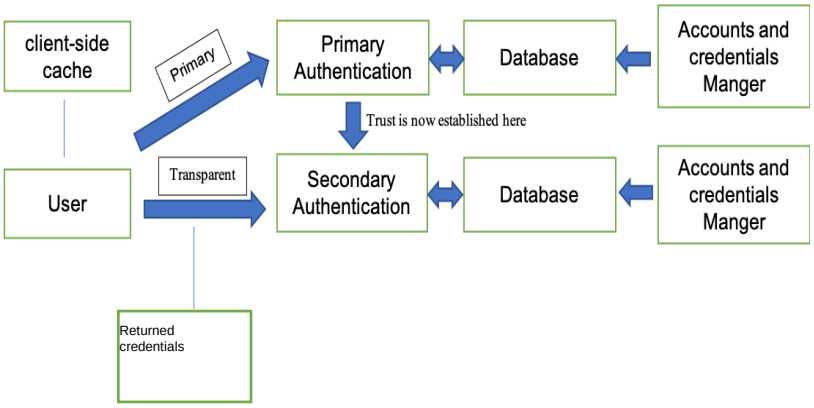

Fig 4: PKI System

2.3 Types of SSO

There are three main variants of SSO: Web SSO, Legacy Web SSO, Federated SSO. We have given a brief description of each of them below [3].

1. Web SSO: Unique web authentication, known as the Web Single Sign-On, allows users to proceed using a Single Sign-On to access a group of web servers that require authentication.

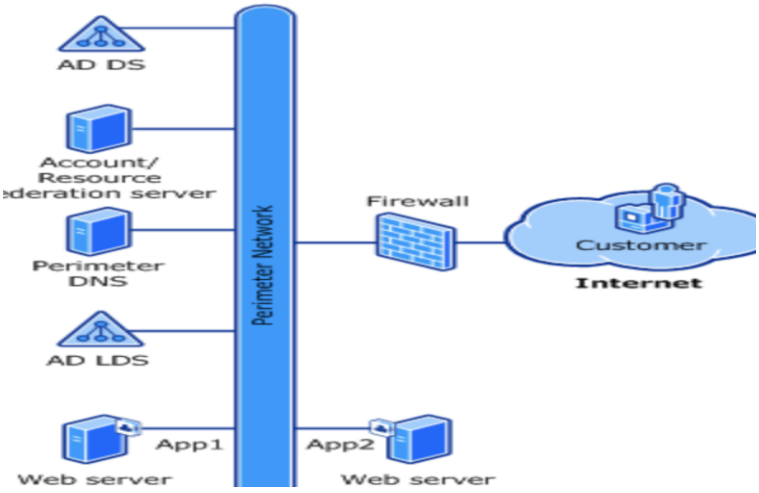

Fig 4:Web SSO

2. Legacy Web SSO: This is also called enterprise SSO. It is very similar to Web SSO with the main difference being in that Web SSO supports for mainly web based applications.
3. Federated SSO: Federation is a relationship which is maintained between organizations. User from each organization gets access across each other's web properties.

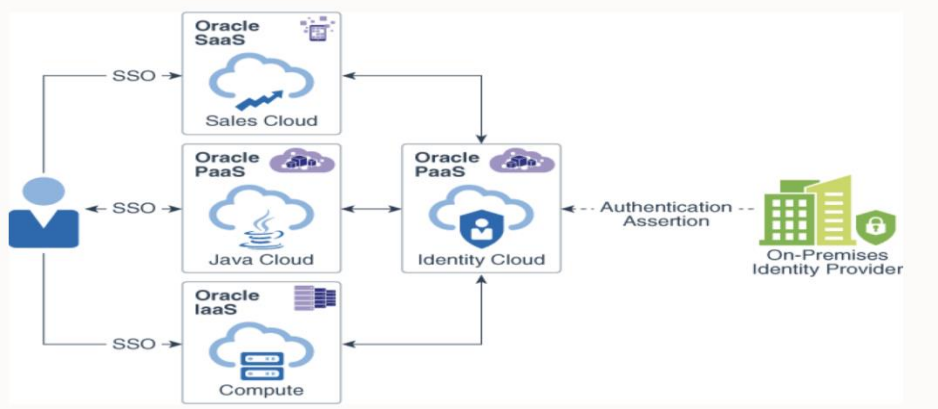

Fig 5: Federated SSO

3. SINGLE SIGN ON ENABLING TECHNOLOGIES AND PROTOCOLS

There are multiple protocols that can be used for SSO implementation. Few of them are listed below [4-6]

1. Lightweight Directory Access Protocol (LDAP):LDAP has primarily been used to authenticate user access to legacy systems and applications. More recently, LDAP has also been used to authenticate user access to DevOps tools. Traditional SSO solutions and LDAP are great at connecting users to their respective applications. How- ever, implementing SSO and LDAP has historically been a give and take relationship for IT admins. For one, traditional LDAP setups are on-prem implementations that can be challenging to implement and maintain. Consequently, admins have been forced to invest a lot of time and effort to support them.
2. Agent Scripts: When changes are made in the authentication policies, the scripts that run in order to facilitate synchronization. This can be done via XML or SQL to manipulate databases
3. Cookies: They are used to aid authentication for a given period of time. Once a cookie expires a user will have to re-authenticate themselves, they are used in a HTTP environment
4. Web Security Service: It allows for cross platform communication across different organizations

According to our study, the protocols used in web SSO are:

1. SAML: open standard that allows identity providers (IdP) to pass authorization credentials to service providers (SP).

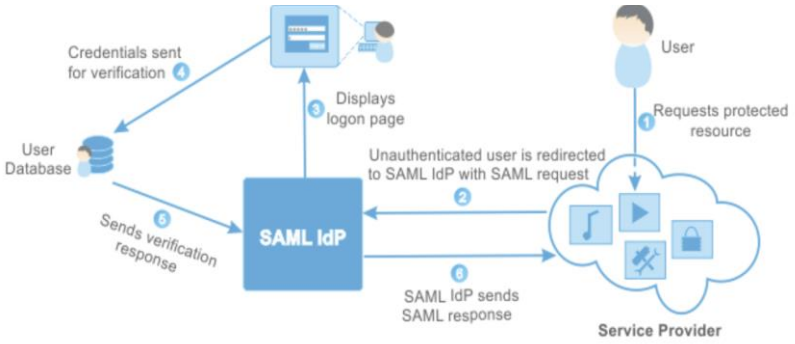

Fig 6: SAML

2. OpenID: OpenID is an open and good user-centric Web SSO solution.

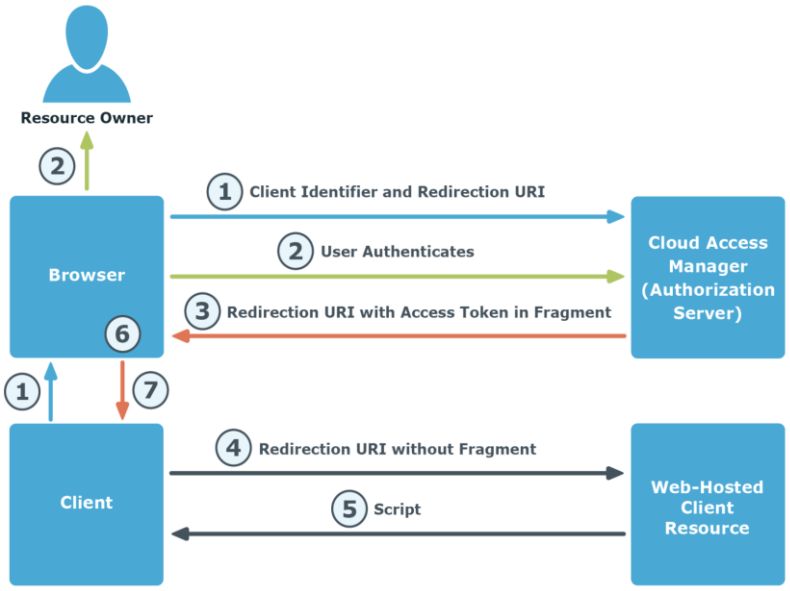

Fig 6: OpenID connect Flow

3. EXPERIMENTAL ANALYSIS

Following the implementation of SSO, a total of over 65,000 logins were sampled systematically during a 7 day period among 2256 end-users. The reduction of login time over the 7 day period showed a net gain of 168.3 h per week. This test is proof of SSO being a large gain both in terms of security and time saved. Time saved is a very big positive for companies as time saved equates to money saved. SSO had a positive impact on efficiency and productivity, and is an effective and cost-effective method to liberate users from repetitive and time consuming logins to software applications.

4. CONCLUSION

SSO is an access control method that allows a user to access multiple domains on a single step of authentication. Thus, relieves the user from the hassle of remembering numerous passwords for multiple applications. SSO is used for user convenience. However, if the main key of the authentication is breached, then the user's crucial details may get compromised. As discussed, this threat of losing master key can be reduced by using SSO with MFA. Also, the analysis proves that there can be a significant increase in efficiency with the help of SSO as an authentication technique.

REFERENCES

[1] B.Li, S.Ge ,T.Wo ,and D.f.Ma, "Research and implementation of single sign-on mechanism for asp pattern," vol. 3251, 10 2004, pp. 161–166.

[2] V. Radha and D. Reddy, "A survey on single sign-on techniques," Procedia Technology, vol. 4, p. 134–139, 12 2012.

[3] C. A. Ardagna, E. Damiani, S. De Capitani di Vimercati, F. Frati, and P. Samarati, "Cas An open source single sign-on solution for secure e-services," in Security and Privacy in Dynamic Environments, S. Fischer- u bner, . Rannenberg, Lngstro m, and S. Lindskog, Eds. oston, A Springer US, 2006, pp. 208–220.

[4] M.- . D'Costa-Alphonso and . Lane, "The adoption of single sign-on and multifactor authentication in organisations: A critical evaluation using toe framework," Issues in Informing Science and Information Technology, vol. 7, 05 2010.

[5] S.-T. Sun, E. Pospisil, I. Muslukhov, N. Dindar, K. Hawkey, and K. Beznosov, "What makes users refuse web single sign-

on? an empirical investigation of openid," in Proceedings of the Seventh Symposium on Usable Privacy and Security, ser. SOUPS '11. New ork, N , USA Association for Computing Machinery, 2011.

[6] D. Dasgupta, A. Roy, and A. Nag, Multi-Factor Authentication. Cham: Springer International Publishing, 2017, pp. 185–233.