International Journal of Modern Electronics and Communication Engineering (IJMECE) ISSN: 2321-2152
Volume No.-6, Issue No.-6, November, 2018

Single Sign-On Enabling Technologies and Protocols

R.Rackymuthu [1]
Department of Computer Science
Government Arts College
Coimbatore, India
E-mail: rackymuthu@yandex.com

V.B.Buvaneswari [2]
Department of Computer Science
Government Arts College
Coimbatore, India
E-mail: buvann@gmail.com

Abstract: In today's digital era, users are increasingly using a huge number of applications **every** day. For accessing services, the users first authenticate themselves and need to maintain a separate username and password for every application. Single sign-on (SSO) is a mechanism for using a single action of authentication to permit an authorized user to access all client application. The Single Sign-On system generates authentication information accepted by the various applications and systems. SSO client applications are without being prompted to log in again at each client application during a particular session. Single Sign-On reduces the risk for the administrators to manage users centrally. Single Sign-On allows users to access multiple services or applications after being authenticated just once.

Keywords: Single Sign-On, OIDC, SAML, Authorization Code Flow, Implicit Flow, Hybrid Flow.

I. INTRODUCTION

Today's internet world, Application service Provider is provides a standard interface to a countless users and also a standard connection point to various application providers. As every application has its security and user information are not correlated making the user management complicated and unsafe. In order to address this kind of issues to the user convenience and security, the commonly used technique is Single Sign-On (SSO). Single Sign-On is an access control method which asks a user to log in once and without any further login criteria, user is allowed to access the resources of multiple software systems securely.

Prior to SSO, a user was supposed to login with the new account each time the new application was opened. Hence, was supposed to memorize number of passwords which is really a difficult task to perform. To deal with this, the users usually set a simple and almost the same passwords for every application. This approach is very easy but has a potential threat in security. Choosing simple passwords make a cracker's job very easy.

An attacker can guess the user password and gain access to all of the application and confidential information. With the introduction of SSO, users are being freed from this menace. They just want to authenticate themselves once and then can easily access the multiple applications.

II. SINGLE SIGN-ON (SSO)

A. Overview

The single sign-on is an access control method. This allows to user when trying to login the application. SSO is authenticates the user and allows the user to access the application.

Figure1. Single Sign-On

After the successful logins, the user can securely access all other SSO client applications without entering the user credentials until the user manually log out or the user session is expired.

Figure1 shows a property, where a user logs in with a single ID and password to gain access to a connected application or application without using different usernames or passwords, or in some configurations seamlessly sign on at each application. This application can be within a single organization or different organizations.

B. SSO Architecture

Different types of Single Sign-On 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 and Token-based SSO.

Secure Client-Side Credential Caching and Secure Server-Side Credential Caching come under SSO with multiple sets of credentials. Public Key Infrastructure based SSO and Token-based SSO come under SSO with a single set

of credentials. Depending on the properties and usage, these architectures can be applied to various kinds of situations.

SSO with Single Set of Credentials: Single Sign-On service provides the management of services, implement the Single Sign-On with single set of credentials. The feature of this SSO architecture is that it is well suited for a homogenous environment where single naming account format and same authentication protocols are supported and identified by every entity in the whole network system.

C. Literature Survey

The last few year security is the major issues and to make it more robust against the unauthorized access is the major process. Single Sign-On is one of such mechanism which provides higher security for users.

Author Manoj V. Thomas and Anand Dhole, K. Chandrasekaran are discussed in "Single Sign-On in Cloud Federation using CloudSim" about the existing single sign-on (SSO) systems such as OpenID and OAuth for authenticating the web sites. They consider the web sites as relying parties (RPs) and provides effective user authentication as a service to identity providers (IdPs) likes Google or Facebook etc. The author goals first privacy-respecting Single Sign-On system for the web, called SPRESSO (for Secure Privacy- Respecting Single Sign-On). This system is easy-to-use, decentralized, and the platform independent. The author also explains the formal analysis of SPRESSO based on an expressive model of the web in order to formally prove that SPRESSO enjoys strong authentication and privacy properties.

Author Prashant Kumar Gajar, Arnab Ghosh And Shashikant Rai is discussed in “Security Risks And Mitigating Strategies” about a password -authenticated key agreement scheme using smart cards. In terms of efficiency, besides the low costs, proposed solution builds on the efficient cryptographic for smart card environment. It also produce effective results while analyzing mutual authentication, key agreement, initiator anonymity, and the functionality of password updating, Denial of Service (DoS) attack prevention and initiator traceability.

Author Arul Princy is discussed in "A Survey on Single Sign-On Mechanism for Multiple Service Authentications” about specifically focused on developing the SSO for distributed environment by giving a detailed survey. The author founds that most existing terms cannot preserve user anonymity when possible attacks occur and those schemes are insecure. Also in existing Single Sign-On schemes have not been formally proved to satisfy credential privacy and soundness of credential based authentication. To overcome this issue, security model of single sign-on scheme with authenticated key exchange. Specially, difference between soundness and credential privacy is pointed out and they define them together in one definition. Also, propose a provably secure single sign-on authentication scheme

Author Kirstie Hawkey, Konstantin Beznosov, is discussed in " Investigating User’s Perspective Of Web Single SignOn:Conceptual Gaps” about detailed analysis the Single Sign-On enabled user accounts is performed on different web applications to get the users interest and found that user’s perception of web SSO is still poorly understood. After finding the issues the author offers a web SSO technology acceptance model with design improvements. To reduce user’s privacy concerns, it is crucial that relying parties practice the principle of gradual engagement, and IdPs provide fine-grained privacy control and on-login profile switching option. In addition, future research must investigate about how to enhance users’ security perception and mitigate IdP.

III. SINGLE SIGN-ON PROTOCOLS

OIDC (OpenID Connect) and SAML (Security Assertion Mark-up Language) are the most widely used federation protocols for web-based single sign-on.

A. SAML

SAML builds the hopeful relation on a digital signature, SAML tokens issued by the identity provider are signed XMLs, the application validates the signature itself and the certificate it presents. The user information is included in a SAML token, among other information.

B.OIDC

OIDC define three authentication flows.

1. Authorization Code Flow
2. Implicit Flow
3. Hybrid Flow

The Authorization code flow and implicit flow based on the OAuth flow. The main difference is OIDC is also issuing the ID token.

1. Authorization Code Flow

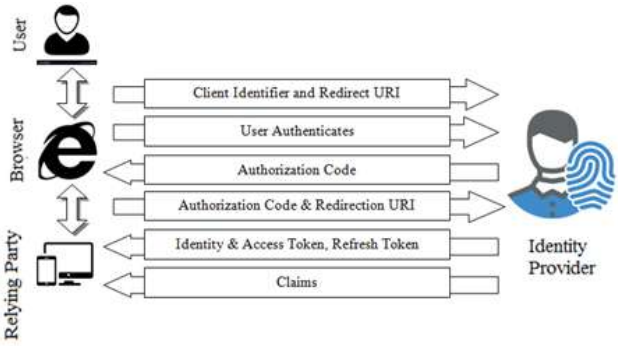

Figure2. Authorization code flow

Figure2 explains about the steps authorization code flow.

(a) Client identifier and & Redirect URI

In this flow, user authenticate the application by clicking the log in its redirect to user agent typically browser. User-agent sends a request to the identity provider with some data.

Figure3. Redirection URI of the Authorization Code flow

The Host indicates the location of the identity provider authorization endpoint. Response type indicates the authorization code flow. OIDC in Scope is indicating the request for opened connect authentication, and the client needs to access the email and profile information of the user. Client ID of the relying party is registered with the identity provider. The state is a value set by the relying party to maintain state between request and the callback itself. Finally, redirect URI indicates the callback location once the user authenticated. These properties are explained in Figure3.

(b) User Authenticate

The user is redirected to the identity provider login screen where they enter login credentials. Identity provider indicates the user and asks for consent to the relying parties request to access their identity. If consent given the identity provider sends an authorization response message from its authorization endpoint. It is redirects to the client by using the redirect URI.

(c) Authorization code & Rediret URI

Figure4 shows the relying party makes a request for ID the token to token endpoint. By sending with base64 encoded client ID, secret key, grant type, and redirection URI.

Figure4. Authorization base64 code

The identity provider authenticates the client ID and secret. And validate the authorization code and redirect URI.

(d) Identity token, Access token & Optional Refresh Token

If the authorization code is valid, the identity provider sends the response back with a request to relying party with the id token, access token and optional refresh token. The client validates the id token and if it is successful the identity is proven.

This is access token and id token is base64 encoded JWT format. This access token contains the user information such as email id and profile information of the user.

2. Implicit flow

Implicit flow contains two types of authentication.

2.1 Implicit flow with ID token

The client application request only the id token. And request any additional user claims are sent within the ID token itself. Implicit flow redirects URI shown in figure 5.

(a) Client identifier & redirection URI

In this type, the user clicking login the client application is redirected to identity provider with response type id token.

Figure5. Redirect URI of Implicit flow

The response type id token is indicates the implicit flow and nonse value used to associate a client session with an id token, and to mitigate replay attack.

(b) User Authenticates

The user logs in consent to access their identity is requested. when the user gives consent, A authorization response message is sent from the authorization endpoint, which redirect the user agent back to the client.

(c) Redirect URI with the ID token in Fragment

The redirection URI includes the ID token in a URI fragment. Figure6 shown the redirect URI contains ID token contain the standard claims, along with the profile and email.

Figure6. Extract access token with user data

(d) Redirect URL without fragment

The user agent follows the redirect URI, but retains the ID token. The applications retain the web page that contains the script.

(e) Script

The script can extract the ID token from the full redirect URI. The user agent executes the provided script, and passes the extracted ID token to the client application. The client application validates the ID token, and now has proof of identity and all the claims require.

2.2 Implicit flow with ID token and Access token

The client application requests both an ID token and access token. The access token is used to request additional claims from the user info endpoint. OIDC implicit flow is shown in Figure 7.

This type is almost similar to id token except the initial redirect. And use the token with id token in the response type.

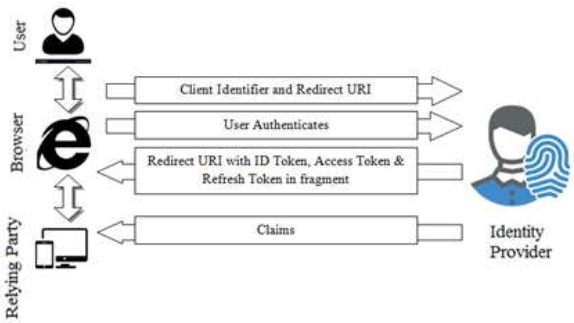

Figure7. Implicit flow with the access token and ID token.

After the user successful authentication and consent the identity provider redirect to user agent. The redirect response contains the access token with the ID token in redirection URI fragment. The client application extracts the ID token, and access token and validate the ID token. Because the ID token does not contain the user profile claims. The access token is sent to the user info endpoint. Once the token is validated, the claims are returned to the application.

3. Hybrid Flow

The hybrid flow is a combination of the authorization code and implicit flow. The tokens can be returned by both authorization and token endpoint. Where the tokens are returned from, depends on the response type specified in the redirect URI. The response type can be set either Code Token, Code id token or Code id token token. Hybrid flow response type is shown in figure 8.

```
response_type=code%20id_token%20token
```

Figure9. Multiple response code in hybrid flow

For example, the client application choose code ID token, the user agent redirect to the identity provider passing the response type code ID token along with the scope it requires. The end user logs in and asks for consent. After the consent is given, an authorization message is sent, which redirect the user agent back to the client applications.

Once the authorization token is validated, the access token is sent back, along with the id token. This id token is compared to the previous one for validation.

IV. BENEFITS AND DRAWBACKS OF SSO

The implementation of SSO has both positive as well as negative impact. The pros and cons of SSO are discussed below

A. Benefits of Using SSO

1. Increased User Productivity

With the advent of SSO, users no need to memorize multiple IDs and passwords for login the applications. Thus, SSO truly burden of users by eradicating the hassle of multiple passwords. The users just need to go through login in single step and enjoy access to multiple applications.

2. Increased developer productivity

The implementation of SSO provides a monotonous authentication framework to the developers. The developers need not worry about authentication at all if the SSO mechanism is independent. When once user login single application, the user can login all other application without entering the username and password.

3. Simple administration

The administration burden of user account management is also simplified when applications participate in the SSO protocol. As Single Sign-On is only deals with authentication, the level of simplification depends merely on the applications. Thus, the some user specific requirements may still be required to be set up by the applications.

B. Drawbacks of Using SSO

1. Scalability problem

The SSO implementation can be difficult, time consuming and expensive to fit into existing applications.

2. Logged in desktops

Although the SSO implementation reduces security risks the threats can be manifold. For instance, a legitimate user might sometimes just walk away from his system leaving user account logged in. A malicious user can easily gain access to it and hence all the authorized resources are compromised. Although, this problem can encounter with security generally, the after effects with SSO is worse as without SSO only one resource gets compromised.

3. Single point of failure

The arrangement is prone to denial of service (DOS) attack as the authentication mechanism is centralized.

C. Challenges in the Growth of SSO

There are many challenges in the growth of SSO. Few of them are as follows

1. Resistance to change

Most of the users are comfortable with the existing system of multiple login system and don't want to switch from the traditional system to a new method of single step authentication. They mostly prefer to users also highlight puse the password manager feature in the browsers.

2. Security issues

Most of the people impressed to using single sign-on mechanism, they are providing directly their usernames and passwords to the server and hence, their sensitive information is stored locally somewhere.

3. Phishing issues

The users also phishing attacks as one of the main reasons that hinder the SSO adoption as they couldn't really find any distinguished difference between the real websites and the bogus ones.

4. Trust issues

This factor is the most crucial one. The users often hesitate to provide their personal and sensitive information to the websites using SSO. Often the websites using SSO feature ask the users to permit them to access their contact list, location and other sensitive information which clicks every users' mind before user agrees to the clause.

V. COMBINATION OF (SSO) WITH MULTI-FACTOR AUTHENTICATION (MFA)

The process of proving the user identity and verifying that user are having the same entity as user claim to be is referred to authentication. There are a variety of mechanisms available for authentication providing a additional security. these are; Biometrics, One Time Passwords, Digital Signatures etc. The authentication is called as Multi-Factor Authentication (MFA) if at least two of the three authentication mechanisms listed below are satisfied at a particular instance of time by a user.

A. Something you know (password or PIN)

B. Something you have (smart card or a mobile phone)

C. Something you are (represented by, say, a fingerprint)

Figure9 shows the combination of SSO with MFA where a user first signs in via Single Sign-On approach and has to get through second authentication step using MFA like One Time Password (OTP) before getting access to the resource.

An ATM-based transaction is a example of Multi-Factor Authentication. As here two of the above three conditions need to be satisfied for a successful communication. The ATM cardholder must have his/her ATM for a transaction satisfying the type "Something you have" and besides this, he/she must have the pin code of the same which satisfies another condition "Something you know".

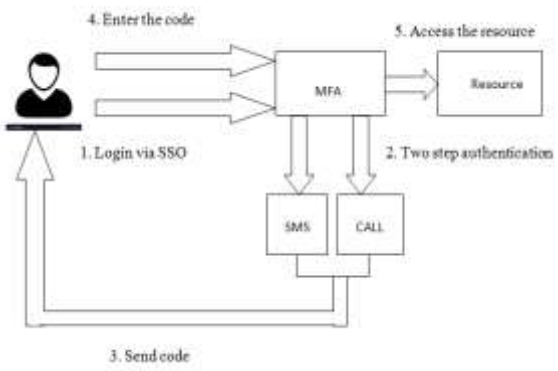

Figure9. Multi-factor authentication in Single Sign-On

Multi-Factor Authentication is used in collaboration with SSO can help to reduce the security issues related to SSO and hence, making it more secure. OTP can be delivered to the user through the SMS, phone call or using some mobile application. The time limit of the OTP is fixed after the session expires making the code invalid.

VI. CONCLUSION

The OpenID Connect is easy to implement compare to SAML. In fact, a minimal working code with the selected flows can be built almost instantly. The hybrid flow in the OIDC flow, it offers more flexibility with the token flow. But this less secure than the authorization code flow because some tokens are exposed to the user agent. Based on the security in the SSO allows a user to access multiple domains on a single step of authentication. Thus, relieves the user from the hassle of remembering huge number of passwords for multiple applications. SSO is used for user convenience. However, if the main key of the authentication is loosed, 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. The combination of SSO and MFA may allow a user to not worry about the negative consequences of losing his/her master key. Due to second step authentication, an intruder cannot access a user's confidential data by acquiring just a key.

REFERENCES

[1] Patil, A., Prof. Pandit, R., and Prof. Patel, S. 2013. Analysis of Single Sign-on for Multiple Web Applications.
[2] Ardagna, C. A., Damiani, E., Vimercati, S. C., Frati, F.and Samarati, P. an Open Source SSO Solution for Secure e-Services.
[3] Sun, E., Muslukhov, S. T., Pospisil, I., Dindar, N., Hawkey, K., and Beznosov, K. What Makes Users Refuse Web Single Sign-On? An Empirical Investigation of OpenID. J. Symposium on Usable Privacy and Security (SOUPS)
[4] Khalid Bashir And Saman Asif"Important Considerations For Single Sign-On Solution"In International Journal Of Multidisciplinary Sciences And Engineering, Vol. 1, No. 1, September 2010
[5] Bernardo Machado David, Anderson C. A. Nascimento, Rafael Tonicelli: "A Framework for Secure Single Sign-On"
[6] Kirstie Hawkey, Konstantin Beznosov, University Of British Columbia" Investigating User's Perspective Of Web Single SignOn:Conceptual Gaps.
[7] Arul Princy "A Survey on Single Sign-On Mechanism for Multiple Service Authentications" IJCSMC, Vol. 2, Issue. 12, December 2013
[8] Prashant Kumar Gajar, Arnab Ghosh And Shashikant Rai "Bring Your Own Device (Byod): Security Risks And Mitigating Strategies "Volume 4, No. 4, April 2013 Journal Of Global-Research In Computer Science
[9] C. Ramakrishnan, S. Dhanabal" Security Analysis of a Single Sign-On Mechanism For Distributed Computer Networks ",IOSR Journal of Computer Engineering.
[10] Yaser Fuad Al-Dubai & Dr. Khamitkar S. D"Kerberos: Secure SSO Authentication Protocol Framework for Cloud Access Control"
[11] Madhavi A. Indalkar , Ram Joshi "Efficient and Secure Single Sign on Mechanism for Distributed Network"- International Journal of Computer Applications (0975 – 8887) Volume 99– No.8, August 2014
[12] Manoj V. Thomas, Anand Dhole, K. Chandrasekaran: "Single Sign-On in Cloud Federation using CloudSim" I. J. Computer Network and Information Security, 2015
[13] Daniel Fett,Ralf Küster,Guido Schmitz "SPRESSO: A Secure, Privacy-Respecting Single Sign-On System for the Web" in ACM 2015
[14]Lawrence O'Gorman" Comparing Passwords, Tokens, and Biometrics for User Authentication "Proceedings of the IEEE, Vol. 91, No. 12, Dec. 2003, pp. 2019-2040 2003 IEEE
[15] Waleed A. Alrodhan. Privacy and Practicality of Identity Management Systems: Academic Overview. VDM Verlag Dr. Müller GmbH, Germany. ISBN 978-3639380255. 2011.
[16] San-Tsai Sun. Towards improving the usability and security of Web single sign-on systems. Ph.D thesis, University of British Columbia. November 2013.

AUTHORS BIOGRAPHIES

First Author

R.Rackymuthu B.Sc., M.Sc., pursuing M.Phil in Department of Computer Science, Government Arts College, Coimbatore-641 018.

Second Author

V.B.Buvaneswari M.Sc., M.Phil., Assistant Professor, Department of Computer Science, Government Arts College (Autonomous), Coimbatore - 641 018.