IJCSN - International Journal of Computer Science and Network, Volume 8, Issue 2, April 2019
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

# Effective Comparison of Single Sign-On Protocols

**[1] R Rackymuthu; [2] V B Buvaneswari**

[1] M.Phil Research Scholar
PG & Research Department of Computer Science, Government Arts College (Autonomous)
Coimbatore-18, Tamilnadu, India

[2] Assistant Professor
PG & Research Department of Computer Science, Government Arts College (Autonomous)
Coimbatore-18, Tamilnadu, India

**Abstract -** In today's digital era, internet and cloud computing, usability and security are most extremely important. Swift interaction on the internet is, however, often compromised by the need of frequently entering usernames and passwords. To prevent users from having to authenticate frequently, services can be provided with Single Sign-On. There are several SSO protocols available, all of which have different properties. This paper provides a comparison of these protocols based on their suitability and security. The comparison should help to decide and build SSO protocol. Information about several SSO protocols are gathered. The protocols are compared based on their properties and functionality namely OpenID Connect, SAML and LDAP. This paper focuses on performing a security analysis on these protocols and recommends a better protocol for implementing a single sign-on service. It turned out that SAML is used most often. Protocol called SAML is a commonly used protocol with security vulnerability in the implementation. But OpenID Connect is the fastest growing protocol. The LDAP protocol was built for server SSO in a local network and not for web applications and also LDAP is the least famous protocol. Both LDAP and SAML are becoming out dated, whereas OpenID Connect is new, web oriented and used by many leading companies such as Google, Yahoo and Facebook. OpenID Connect offers authentication and authorization. It uses modern standards and has a growing community. Because of this OpenID Connect is the best protocol to build SSO.

***Keywords -*** *OIDC, SAML, LDAP, SSO, XML.*

## 1. Introduction

Single Sign-On is a reliable and secure network system which keeps growing in complexity due to the interfaces with multiple user logging sub-systems for different application and to ensure the safety of the network environment for everyone involved. Single Sign-On provides a secure and reliable network in every system. It aims to achieve overall security for cluster of applications. Single Sign-On is shortly called SSO. SSO system's necessary requirements must be identified. Thereafter, user identity management and different authentication mechanisms were defined together with the network protocols and standards to ensure a safe exchange of information within and outside the company. The focus of this study is to provide insight into various SSO protocols that meet the requirements and to compare them. Based on the pros and cons of each protocol, one best protocol will be chosen for which an implementation plan should be made.

### 1.1 Need of SSO

Today's internet world every user uses many systems. Every system has its own user management and authentication subsystem and own security policy. The user accessing different systems has to use different user credentials for independent authentication. In this situation, remembering the username and password become a heavy burden to the users. At the same time, the authenticated user information transfer over the network. This causes a very high-security risk and issue. To fix this issue, single sign-on concepts have come into being. Single Sign-On is a comfortable authorization mechanism to users for using multiple applications. Fig 1 shows the context diagram of SSO. This diagram provides a simple overview of an SSO protocol. This protocol allows multiple users to access computers, tablets, mobile phones etc. These users can also access to a service which can be a data, an application, or a part of an application.

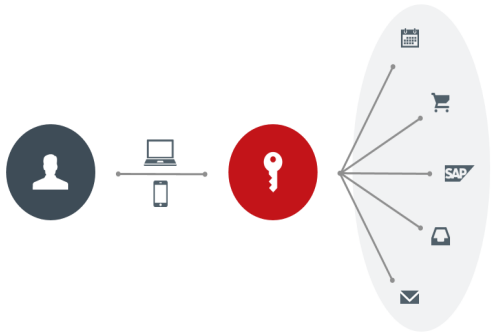

Fig 1: Multiple Web Application Login using single Username and Password

## 2. Literature Review

Security is the major focus and to make it more robust against the unauthorized access is the important concern (Raksha patil et. al.,2016). Single Sign-On is one of such mechanism which provides higher security for users. Replication system is responsible for propagating the data modification made by each member to the rest of the group, and resolving any conflicts that might arise between concurrent changes made by different members. The Lightweight Directory Access Protocol (LDAP) is the open standard which supports the multiple platforms. It simply uses the client Server model for communication. The Lightweight Directory Access Protocol provides a low-overhead mechanism which is used to access the X.500 directory. LDAP help to provide data synchronization over multiple systems by providing the replication technique.

Organizations can easily, yet securely share identity information. Security is improved by eliminating the possibility of shared accounts. User experience is enhanced by avoiding additional usernames and passwords. (Kelly D. lewis et al., 2009). Service providers can use different security protocols, such as signed only, versus signed and encrypted. In addition, some service providers may only use the name ID section of the assertion, while others might use custom attributes only. This upfront documentation can save troubleshooting time during the implementation and testing phases of the project. Furthermore, during testing phases it is helpful to use a sample test site for the service provider and also to test with SAML signed assertions. The sample test site allows isolating a test of SAML connection between the two partners, before testing of the application. Both the identity provider and service provider should utilize metadata files, not only to speed up manual work when entering data into the federation software, but to also reduce human error.

Security analysis is conducted consisting of two parts; first, an analysis of the source code (written in C# language) using the VCG tool that discovered four weaknesses in the code namely unsafe code directive, code that appears to contain a hard-coded password, an application that appears to log a user password and .NET debugging-enabled code. Secondly, an analysis of the HTTP messages' uses OWASP ZAP tools. This discovered two weaknesses namely cross-site scripting (XSS) attacks and remote file include (RFI). (Waleed A. Alrodhan et al., 2017). There are several security issues within OpenID. Some of these issues have been addressed in OpenID latest version OpenID Connect. Although some of these issues remained in OpenID Connect, like the XSS, CSRF and Invalidated/forward redirect vulnerabilities, an effective solution has been proposed by OpenID to tackle this issue. The solution involves the use of a cryptographic hash function and a Token Binding value within tokens. However, other security issues (e.g. log files exposure) are still unresolved. In addition, novel high-level integration model of OpenID Connect would result in number of advantages regarding security, privacy, practicality and scalability.

Security domain of IoT faces challenges regarding authentication and authorization. It contains the most recent research and categorizes it from multiple perspectives. It shows how context-awareness extends security and what approaches exist to incorporate context-awareness into IoT security. It shows how existing and current, widely adopted technologies are adapted for the IoT and survey new security proposals designed specifically for that environment. (Michal et al., 2018) Michal et al. discussed whether security solutions for centralized or distributed architectures are favorable to machine-to-machine or user-to-machine. There is no similar study or survey of IoT security or any other study containing the latest IoT security research. This reason allows them to reapply existing knowledge and deal with the security issues that are preventing IoT popularity and adoption increasingly among end users.

## 3. Single Sign-On Protocols

This research aim to identify the best protocol to implement the SSO service. This is done by comparing the characteristics of SSO protocols and challenges in real-time implementation. The entire Single Sign-On is covered under three techniques. That is authentication, authorization and session management. This is the base of the SSO.
Following protocols are involving in the SSO implementation.

Common Protocols involving in SSO:

1. OpenID Connect (OIDC)
2. SAML
3. LDAP

## 3.1 OIDC

This is the OIDC version of the OAuth2 Authorization Code Grant. This flow has the response type set to "code". Fig 2 shows the flow of the OIDC authorization code flow. Steps involve in the OpenID Connect authorization code flow

1. Authentication of end user
2. Authentication of Relying Party (Client, application), optionally.
3. Relying Party (RP) does not notice end user's credentials (password)
4. The User Agent does not notice the Access Token or ID Token
5. SSO to RP
6. Obtain multiple Access Tokens or multi-scope Access Token.
7. Make API (Resource) call to API Provider (Resource Server) with Access Token.

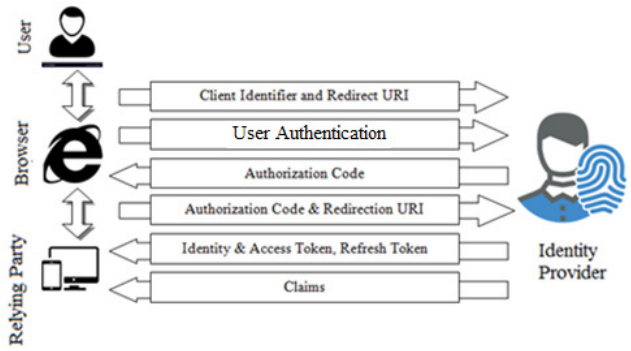


Fig 2.OIDC Authorization code flow

### 3.1.1 Client identifier and & Redirect URI

In Client identifier and redirect URI, the user authenticates the application by clicking the log in which is redirected to user agent (Web browser). User-agent sends a request to the identity provider with some data.

The Host indicates the location of the identity provider's authorization endpoint. Response type indicates the authorization code flow. OIDC scope is indicating the request for OIDC 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. Finally, redirect URI indicates the callback location once the user has authenticated.

### 3.1.2 User Authentication

The user is redirected to the identity provider login screen where they enter login credentials. Identity provider identifies the user and asks for consent to the relying parties to access user identity. After the consent is received, the identity provider sends an authorization response message from its authorization endpoint. It is redirected to the client by using the redirect URI.

### 3.1.3 Authorization code & Redirect URI

The relying party makes a request for ID token to token endpoint along with base64 encoded client ID, secret key, grant type, and redirection URI. The identity provider authenticates the client ID and secret key and validates the authorization code and redirect URI.

### 3.1.4 Identity token, Access token & Optional Refresh Token

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

### 3.1.5 User Login

ID Token contains the user related information. The client application validates the ID token information and allows the user to login.

## 3.2 SAML

SAML is an open standard for exchanging authentication and authorization data between the identity provider and a service provider. This is shown in Fig 3. Steps involve in the SAML are:

1. The protocol starts with a client contacting the service.
2. To access the service, the client sends a message to the Identity Provider (IdP) and Authorization Server (AS) to ask permission to access the application.

3. The IdP/AS server asks the client to provide login credentials.
4. The client returns the credentials to the IdP/AS server which in turn validates them.
5. If the credentials are valid, the IdP contacts the AS and asks for authorization data.
6. The AS returns the access token to the IdP.

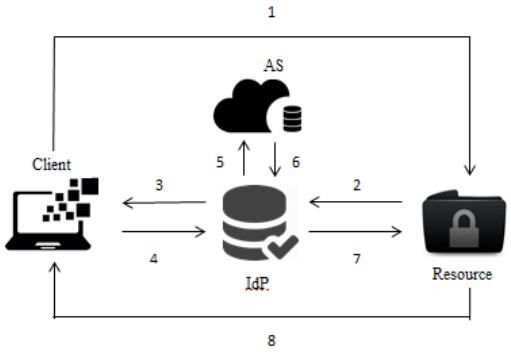


Fig 3: SAML Protocol

7. Which in turn informs the service that the user is authenticated and sends an access token.
8. The service grants access to the requested resource.

## 3.3 LDAP

The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing distributed directory services, which can be used for authentication and authorization. LDAP is often used in combination with an active directory. A directory is a kind of specialized database. A simple example of a directory is a phone book.

LDAP is a broad solution that can be used for many purposes including authentication and authorization. LDAP is a good example of an Enterprise SSO solution. It is robust and supports many features more than the user needs. Because of this, the protocol is harder to understand than SAML or OpenID Connect. LDAP is a suitable solution for users looking for an SSO solution that can work with directories or an enterprise solution that needs to be implemented.
LDAP is based on DAP which is a protocol defined by the X.500 computer network standards. Early adopters of DAP thought that it was fairly complex and not well suited for the desktop computers of that day. DAP is large, complex, and difficult to implement, and most implementations perform poorly. Two independent groups devised similar protocols that were simpler and easier for desktop computers to implement. The two lightweight protocols for desktop computers were called: Directory Assistance Service (DAS) and Directory Interface to X.500 Implemented Efficiently (DIXIE). After DIXIE and DAS showed that a lighter-weight access protocol could be produced forX.500, the members of the OSI-DS Working Group (of the IETF) decided to join forces and produce a new full-featured, lightweight directory access protocol for X.500 directories.

### 3.3.1 LDAP Protocols

The LDAP protocol consists of LDAP clients and an LDAP server. The clients create an LDAP message that contains a certain request and sends it to the server. The server processes this request and sends the results back to the client as one or more LDAP messages. For example, when an LDAP client searches for a specific entry, then it sends an LDAP search request message to the server. Each message contains a unique message-ID generated by the client. The server receives a message from its directory and sends it to the client followed by a separate message that contains the result code. All of the communication between the server and the client is identified by the message-ID provided in the request of the client. If the server searches the directory and finds multiple entries, those entries are sent to the client in a series of LDAP messages, one for each entry, as shown in Fig 4. The search results are terminated with a result code, which contains the overall result of the search operation.

### 3.3.2 LDAP Functionality

These two operations allow user to ask questions to the directory server.
1. Interrogation operations: search, and compare.
2. Update operations: add, delete, modify, modify DN (rename).
3. Authentication and control operations: bind, unbind, abandon.

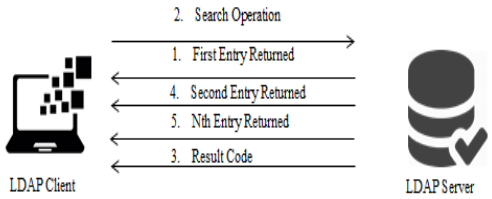


Fig 4: LDAP Communication

The bind operation allows a client to identify itself to the directory by providing identity and authentication credentials; the unbind operation allows the client to terminate a session; and the abandon operation allows a client to indicate that it is no longer interested in the results of an operation it had previously submitted. For distributed authentication, two scenarios in LDAP can be applied:

1.Server 1 might tell Server 2 who the authenticated user is, and Server 2 might simply choose to believe Server 1. This approach requires that Server 2 trusts Server 1 to verify authentication credentials correctly.

OR

1.Server 1 passes the users identity and authentication credentials to Server 2. Server 2 could then independently verify the credentials. This approach requires that Server 1 trusts Server 2 not to misuse the authentication credentials (for example, if the credentials consist of a plaintext password, then Server 2 must not reveal it to a third party).

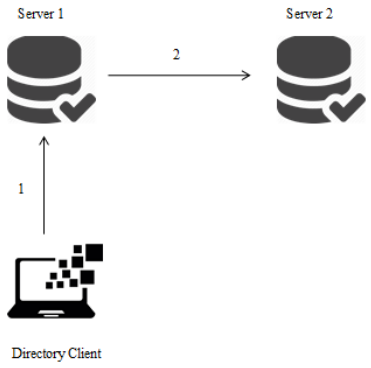


Fig 5: LDAP Authentication

The second scenario can also be executed by using certificates instead of login credentials. If the client has authenticated to Server1 using certificate-based client authentication, Server1 can pass along the digitally signed request and allow Server2 to verify the client's identity directly, since the identity of the client is proven by the digital signature.

Step one in Figure 5 is the same in both scenarios: a user logs in using his login credentials. The second step differs, this can be the message "User X has Logged in" or "User X has logged in and here are the credentials to prove this".

## 4. An Overview of SSO Protocols

Based on this comparison, The OAuth2, CAS and CoSign

Do not meet the requirement of offering authentication and authorization. LDAP is only done authentication and not providing authorization.

Thus, two potential solutions remain: OpenID Connect, SAML 2.0. Because OAuth2 is the basis of OpenID Connect then overview of SSO protocols on shown in Table 1.

**Table 1: Overview of SSO Protocol Comparison**

| Protocol | Auth | Authz | Type | Consent | Expiry |
|---|---|---|---|---|---|
| OIDC | True | True | JSON | True | True |
| SAML 2.0 | True | True | XML | False | True |
| LDAP | True | True | False | False | False |

True = Offers this property

False = Does not offer this property

## 5. Conclusion

OpenID Connect is the newest protocol (2014), and LDAP is the oldest (1993). Because LDAP was released when the internet was still in its infancy, its focus is not on web applications but on communication between servers of an enterprise. SAML and OpenID Connect both offer authentication and authorization, and both focus on the web. This means that they use protocols and notations in their communications that are supported by the web, that they are light weight and that they also support SSO on multiple domains. SAML protocol have major implementation problem named XML Wrapping Attack. That is a chance for the intruders to alter the messages. So OIDC is a best choice for this problem.

SAML is the protocol that is used most often, but OpenID Connect is being adopted rapidly. OpenID Connect can be used for the same things as SAML, but it uses new standards like JSON, REST and more secure than SAML. Also, identity providers that already offer OIDC, such as Google, have started implementing OpenID Connect. This together with the rapid growth of the protocol makes OpenID Connect as platform independent.

## 6. References

[1] Arul Princy (2013) "A Survey on Single Sign-On Mechanism for Multiple Service Authentications" International Journal of Computer Science and Mobile Computing (IJCSMC).

[2] BaranTopal (2016) “Methods of Single Sign-On” KTH, School of Information and Communication Technology (ICT). (CCS).

[3] Carbone, R., Armando, A., Compagna, L., Cuellar, J., and Tobarra, L. (2008) “Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-Based Single Sign-On for Google Apps”. Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering.

[4] Collan, J.,( 2009) “Secure Authentication and Authorization Portal Based on Single Sign-on”. Helsinki University of Technology.

[5] Causton, R. P., (2002) “Smart Card Usage for Authentication In Web Single Sign-On Systems.” Master of Science degree thesis.Helsinki University of Technology.

[6] Daniel Fett,RalfKüster,Guido Schmitz (2015) "A Secure, Privacy-Respecting Single Sign-On System for the Web" ACM.

[7] David, R., Laurie, R., and Chris, M. (2012) “OpenID: the Definitive Guide”. Oreilly Associates Inc.

[8] Fang Yinglan, JinHao and Han Bing (2014) “Single Sign-On Research and Expansion Based On CAS” The Open Cybernetics &Systemics Journal.

[9] Kaufman, C., Perlman, R. and Speciner, M.,(2002). “Network Security Private Communication in a Public World”. 2nd ed. New Jersey: Prentice Hall.

[10] Khalid Bashir And Saman Asif (2010) "Important Considerations For Single Sign-On Solution" International Journal Of Multidisciplinary ciences And Engineering.

[11] Lawrence O’Gorman (2003) "Comparing Passwords, Tokens, and Biometrics for User Authentication " Proceedings of the IEEE.

[12] Manoj V. Thomas, Anand Dhole, K. Chandrasekaran (2015) "Single Sign-On in Cloud Federation using CloudSim" I. J. Computer Network and Information Security,

[13] Madhavi A. Indalkar , Ram Joshi (2014) "Efficient and Secure Single Sign on Mechanism for Distributed Network” International Journal of Computer Applications.

[14] Mohamed Watfa, Shakir Khan, Ali Radmehr (2014) “Implications of SSO solutions on cloud applications” University of Wollongong in Dubai – Papers.

[15] Mary OdilyaTeena.A, Dr.Aaramuthan.M(2017) “Federated Cloud Identity Management: A Study on PrivacyTactics, Tools and Technologies” IOSR Journal of Computer Engineering.

[16] Marise-Marie, Michael Lane (2010) “The Adoption of Single Sign-On and Multifactor Authentication in Organizations – A Critical Evaluation Using TOE Framework” Issues in Informing Science and Information Technology.

[17] Michal Trnka and Tomas Cerny and Nathaniel Stickney (2018) "Survey of Authentication and Authorization for the Internet of Things", Security and Communication Networks, Hindawi.

[18] Prashant Kumar Gajar, Arnab Ghosh And Shashikant Rai (2013) "Bring Your Own Device (Byod): Security Risks And Mitigating Strategies“ Journal Of Global-Research In Computer Science

[19] Pospisil S. T., Beznosov I., Muslukhov, Dindar, N., Hawkey, K.(2011) “What Makes Users Refuse Web Single Sign-On”. Symposium on Usable Privacy and Security (SOUPS).

[20] Raksha patil and Madhuri zawar (2016) "Lightweight Directory Access Protocol for Replication of directory server data", International Journal of Advances in Electronics and Computer Science,

[21] Tian-yuWo, Bo Li, Sheng Ge, and Dian-fu Ma (2015) “Research and Implementation of Single Sign-On Mechanism for ASP Pattern” Computer Institute, BeiHang.

[22] Waleed A. Alrodhan and Alya I. Alqarni (2017), "Security Investigation and Analysis of OpenID: Problems and Enhancements", IJCSNS.

## 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 (Educational Qualification)** Assistant Professor, Department of Computer Science, Government Arts College (Autonomous), Coimbatore - 641 018.