Single Sign on

Anurag Dey,
MSc. (I.T) Final Year Student,
Jain University,
Bangalore- 100, India

Dr. Suchitra Suriya,
H.O.D, MSc. (I.T) Department,
Jain University,
Bangalore-69, India.

***Abstract*— Today almost everywhere internet is available. Everyone is dealing with the applications available on internet. All applications are hosted on internet is available for everyone anyone can access and make use of it. But some application need some security and only an authorized person can access it. For that those application are having login page where a user have to provide login credentials for getting access to the resources. But during login process the user details is send over the internet to the server to authenticate and during that process a middleman (hacker) can access user detail and make use of it. For that encryption of data is being used and later HTTPS technology came which gives more security for user data. An organization can have multiple applications of different purpose running on servers. Each application may have its own authentication process and force users to remember different authentication credentials (usernames and passwords) for each application. User faced with the problem of remembering multiple credentials, they reuse the same passwords for every application, pick weak passwords, or keep a list of all usernames and passwords is a phone or as a diary entry. Managing multiple authentication credentials is annoying for users and weakens security for the authentication system. SSO (Single Sign On) helps the users in this problem by allowing single username and password to be used and get authenticated once and can able to access any authorized applications. Single Sign On is an access control technique of multiple but independent software systems. With this technique a user logs into the system once and will be access multiple applications without being prompted to login for each and every application. It helps to both admin and user to manage the login credentials very easily and it also gives better security.**

Keywords—Single Sign On, Authentication, Identity Provider, Service Provider, LDAP, OpenID.

INTRODUCTION

Single Sign On (SSO) is a process of accessing multiple secure applications or resources provided by some organization in a single login [7]. User doesn't have to re-enter the username, password and other login credentials for each application. With a single authentication user can access multiple application to which user is authorized to. This is achieved by using a lightweight directory access protocol (LDAP) and store LDAP to SSO servers [4]. Typically this database contains the user login details and roles and other user authorization details. Each application might have different authentication technique, SSO internally translate the application authentication technique and store login credentials accordingly. SSO is different from OAuth, OpenID, Facebook connect, these all requires users to enter login details each time user access the application. But in SSO user is able to access multiple application with single login.

SSO authentication technique is fully dependent on ticket that is being generated when user logs in and that ticket is used as an entry ticket to get authenticated by the applications. There are different approaches through which SSO can be configured like Kerberos based, smart card based, SAML (Security Assertion Markup Language) based etc [6].

Developing own SSO server is a very risky part so there are many vendors that provide SSO server, some of them are open source which can be used by any one and can be modified on their own requirement. Some open source SSO providers are CAS by Jasig, Barebones SSO by Kaseya, Keycloak by Red Hat, WSO2 Identity Server by WSO2 and many others. We will go with CAS provided by Jasig (Java in Administration Special Interest Group). CAS is a SSO protocol for web. Its main purpose is to provide users to access multiple applications with single login. CAS protocol involves minimum of 3 parties a client a web application and CAS server. It may also include some backend services like database. When the client first time visit the web application, the application redirect the client to CAS server Authentication page to authenticate the client, after client successfully enter the username and password the CAS server redirect the client to the web application with a security token, on receiving the security token the application contact CAS server to authenticate weather the ticket is valid or not and after successful authentication the client is given access permission to the services of the application. CAS support many authentication protocols like CAS (version 1, 2, 3), SAML 1.1, OpenID, OAuth (1.0, 2.0) [3].

RELATED WORK

There are many organizations working on Single Sign on (SSO) to make it better and more secure. As security is the major concern in providing tickets and in authentication. These are several researches going on how to provide better security to single sign on authentication techniques.

In March 2012 there was a research paper that reported a field study on security of SSO systems in social login

mechanisms. The authors found 8 very serious logic flaws errors in high end profile ID provided by different parties like Facebook, OpenID, Farmville , Jan rain etc [9]. They did a different case study on actual web traffic that goes through the browser, and they tries to get access to the traffic and get useful data. For that they have used one algorithm to get access to the traffic and recover the important identity information's.

And during that case study the found 8 potential serious logic flows that allows them to get authenticated to the application.

Every flaw allows them to get sign in into victim's user [9].

In May 2014, a vulnerability named Covert Redirect was disclosed related to OAuth 2.0 and OpenID by the Wang Jing, a mathematical PhD student from Nanyang Technological University, Singapore. He discovers serious flaws in 2 widely used security standards i.e. OAuth 2.0 and OpenID could give access to anyone to your account information like Google, Microsoft, twitter, Facebook and many other online services. Bothe the standards are widely used on internet that let users to log into website using these login credentials, user doesn't have to create account for the website [8]. They can use Google, Microsoft, twitter, Facebook login id's to login into other website. Attackers can exploit the flaw to disguise and launch a phishing attack from other websites to get the user details.

Wang Jing called this as a flaw's finder. In this Attacker uses a real website by corrupting the real website login page with the malicious login page and whenever user logs in the attackers get the details of users. It was a very serious problem almost all single sign on protocols got affected [8].

There was a scheme proposed by the Chang and Lee on SSO scheme and claimed better security by providing well organized security arguments. Later in one research it was proved that their proposed scheme was actually insecure and it fails to meet the proper authentication. In demonstrate they did 2 attacks. In first attack they communicated with the legal users twice and able to get their login credentials and then use those credentials to get access to the actual service provider. In second attack the outsider means a non-legal user is able to get access to the service provider [1].

There are basically 3 types of single sign on: web SSO, legacy SSO, federated SSO. Legacy SSO enables users to authenticate and once user is authenticated it creates a trust and grants user to access all web resources. Legacy SSO is almost same as web SSO, where it extends the functionality of SSO to legacy applications and network resources while web SSO only manages web based services. Federated SSO is also much like web SSO but has very much broader concept. It uses SOAP (Simple object access protocol) and SAML (Security Assertion markup language) to enable user to sign in once and able to access multiple applications. SSO can be configured with many technologies [6].

Kerberos Based: It is an authentication protocol developed by MIT which works on the basis of "tickets" to allow user to communicate over non secure network to prove user identity to one another in a secure manner. It is primarily focused onto client server model where both client and server verify each other's identity. It is built on symmetric key cryptography. It includes 3 different parties Client, Key Distribution Center (KDC) and any service server where KDC is fully responsible for authentication and authorization of user and granting tickets to the user for different service [2].

Smart Card Based: In this smart card is used authenticate the user. Smart card is any pocket sized card embedded with integrated chips. Smart cards can provide identity documentation, authentication, data storage, and application processing. Additional software applications also use the smart card, without prompting the user to re-enter credentials.

Smart card-based single sign-on can either use certificates or passwords stored on the smart card [10].

Integrated Windows Authentication: It is an authentication protocol provided by the Microsoft windows. It is commonly used to refer to the automatically authenticated connections between Microsoft Internet Information Services and Internet Explorer. IWA works with most modern web browsers but does not work over some HTTP proxy servers, therefore it is best to use for Intranets where the entire client are on single domain. It supports internet explorer 2, Mozilla Firefox, opera 9.01 and later versions, Google chrome 8.0 and later, Safari [2].

Security Assertion Markup Language: Security assertion markup language commonly known as SAML is an extensible markup language standard that allows user to log on once for different applications. SAML is mainly designed for business to business (B2B) and business to consumer (B2C) transactions. It specifies 3 components

1. Assertions: An assertion is a package of information that supplies zero or more statements made by a SAML authority. SAML assertions are usually made about a subject, represented by the <Subject> element. An important type of SAML assertion is the so-called "bearer" assertion used to facilitate Web Browser SSO. There are 3 types assertions i.e. authentication, attribute and authorization. Authentication validates the user's identity attribute contains information about user. Authorization identifies what the user is authorized to. Assertion is also known as security token
2. Protocol: It defines how a SAML request for assertions and also how it receives it. There are many protocols specified in SAM Core. But in that the most important protocols are the Authentication Request Protocol and Artifact Resolution Protocol.
3. Binding: It defines how SAML messages are being exchanged or mapped to SOAP. SAML2.0 support many binding like SAML SOAP binding, reverse SOAP(PAOS) binding, HTTP redirect binding, HTTP POST binding, HTTP Artifact Binding etc [6].

It is an XML based protocol that uses security tokens contains assertions to pass the user information called as principal between the identity provider and the service provider. The most important requirement that SAML address is web browser SSO (single sign on). SAML defines 3 roles principal (user), identity provider and

service provider. Principal request service from service provider, service provider redirects principal to identity provider for authentication and the user is redirected to service requested for if authentication is successful [6].

SAML has many advantages.

SAML is a standard based protocol which provides interoperability across identity providers and gives freedom to the enterprise to choose their vendor. One click access, password elimination, renewing session automatically makes users life very much easier [6].

It uses strong digital signature to authenticate which gives very great security to application. It is very much fast and light weight to travel across network [6].

The are many companies working on Single sign on some of them are: Janrain, Loginradius, Gigya, Oneall, Onelogin, Ping Identity etc. and also there are many organization has implemented the SSO and also providing for public use some of them are open source and some are Proprietary e.g. Active Directory Federation Services by Microsoft, CAS by Jasig, JBoss SSO by Red Hat, Kerberos by MIT, OneLogin by OneLogin Inc [9], OPENAM by ForgeRock, SAML by OASIS and many more [6].

DESIGN

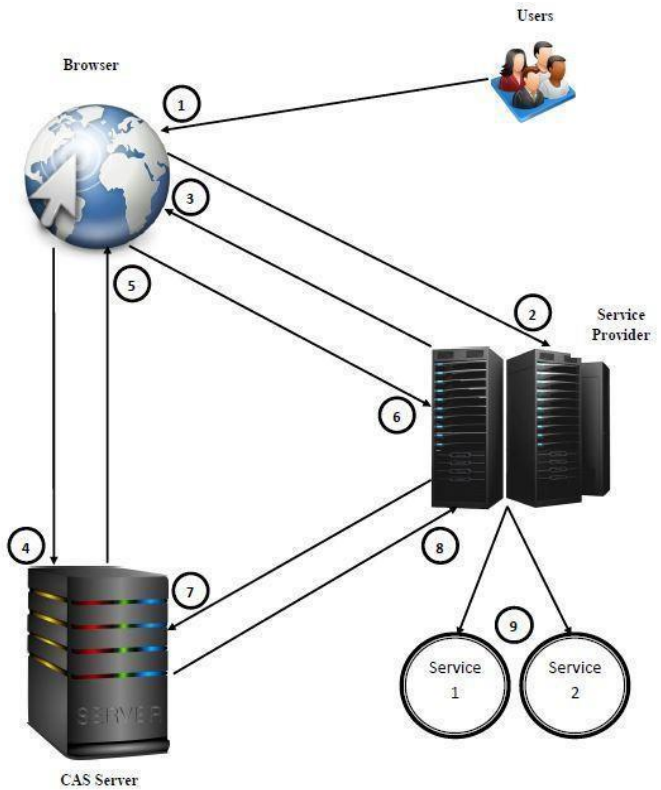

Figure 1: Working of CAS. [3]

1. User enters the URL of service to which user wants to access.

E.g. http://www.ServiceProvider.com/MyApp [3]

2. Service provider checks if user is having valid ticket information or is user is authenticated and authorized to. If user doesn't have ticket information, it means user is not authenticated [3].

3. Service Provider send redirect URL to user browser and browser redirects to that URL

e.g. https://www.CASServiceProvider.com/cas/login 4. User get CAS login page where user enter the login credentials. And if user is authenticated successfully then one security ticket is generated by CAS server and bind with the original service URL which the user was trying to access. E.g. http://www.ServiceProvider.com/MyApp?ticket.... [3]

5. Then CAS server sends the redirect URL with security token to user browser.
E.g. http://www.ServiceProvider.com/MyApp?ticket....

User browser redirect again to service provider with the security token given by the CAS server [3].

6. When service provider receives the ticket it checks the ticket if it is a valid ticket or not. For that it sends the ticket to CAS server to validate E.g. https://www.CASServiceProvider.com/cas/validate?ticket... [3]

7. CAS server receives the ticket and it validates the ticket by checking all its details and send status of ticket to service provider [3].

8. If status send by CAS server is valid then user is able to access any application provided by service provider without login. The ticket generated by CAS is used as an entry ticket for the applications [3].

Each ticket generated by the CAS server has a validity period that expires after the validity is over. Ticket contains the user information like username, role and other information like validity, serial no; ticket no etc which is secured by CAS key. CAS server is a third party which authenticates the user. It does not have much information about the user, only the username and their password is available with CAS server [3]. The main information is available with service provider. When user tries to access any application provided by a service provider then first service provider checks weather user is having proper valid ticket basically tickets are attached with the HTTP URL itself [5]. If user does not have then service provider redirects the user to SSO login page i.e. CAS login page where user is authenticated and after successful authentication CAS provides a valid ticket to user and redirect back to service provider. When CAS generates a ticket then user makes a copy of it into its cache memory. When user reaches the service provider then again service provider checks weather the user is having ticket. If user

has then service provider checks the ticket validity by contacting CAS server and when CAS replies that ticket is valid then user is allowed to use the application or service. And next time again if user visits the application then user sends a copy of ticket stored in cache to service provider and service provider checks the ticket with CAS and allows the user to access the service [3]. This process repeats again and again for every application user tries to access. By this user is skipped with multiple logins of different applications.

CONCLUSION

The document describes the generic working mechanisms of SSO (Single sign on) and gives an overview of how SSO works using CAS as server for authentication. It also gives an idea how user is authenticated and able to access multiple application using a single login. SSO system for web application helps users to save time and also ease frustrations with having to remember the different login credentials for different application by which it reduces the chance of forgetting the username and password. It also reduces the admin work with managing so many login credentials for different users of different applications. SSO is successful in reducing the time and IT cost for managing users login credentials [7]. With greater security level SSO comes with advanced security technique like SAML 2.0 that gives user data an extra security layer. The CAS server provides very much and easy way to setup the SSO system for web application with some configuration. On the other hand CAS doesn't support SAML 2.0 which fails to provide much security for web application, CAS is typically made for small organizations SSO, also SSO is a very critical protocol, if SSO server is off or crashed then user would lose all the access to the applications. Managing SSO server is a also a very critical part, taking backups for database, securing the database with a very much tight security protocol so no outsider can be able to access it. If someone comes to know the username and password then can access all application and that might contain very critical data, so it should be recommended for users to have strong passwords and should never give to others [3].

Overall SSO system is good and bad based on the requirements. Somewhere it lacks on managing the SSO server, taking backup, securing the server database etc and somewhere it gives one login features with great security on web application thanks to SAML 2.0 that provides great security for user authentication. There are many technologies that overcome the problems SSO and today many organizations are moving toward implementing the SSO into their organization [6].

REFERENCES

[1] Mularien Peter. Spring Security 3. BIRMINGHAM – MUMBAI: Packt Publishing, May 2010.
[2] Mularien Peter and Winch Robert. Spring Security 3.1. BIRMINGHAM – MUMBAI: Packt Publishing, May 2012.
[3] "CAS Single-Sign-On for Web". *CAS Protocol* 4 Apr 2014. Apereo Foundation. 23 Jan. 2015 <http://jasig.github.io/cas/4.0.x/protocol/CASProtocol.html>.
[4] "Single-Sign-On". *Single sign-on* 11 Mar 2014. encyclopedia. 23 Jan. 2015 <http://en.wikipedia.org/wiki/Single_sign-on>.
[5] Heaton. Robert. "HTTPS". *How does HTTPS actually work?* 27 Mar 2014. 24 Jan. 2015 <http://robertheaton.com/2014/03/27/how-doeshttps-actually-work/>.
[6] Wulff. Oliver. "Open Source and SOA, ESB and Security". SAML tokens and WS-Trust Security Token Service (STS). 27 Feb 2012. 24 Jan. 2015 <http://owulff.blogspot.in/2012/02/saml-tokens-and-ws-trustsecurity-token.html>.
[7] James. Shakir. "Single Sign On". Web Single Sign-On Systems. Dec 2007. 24 Jan. 2015 <http://www.cse.wustl.edu/~jain/cse57107/ftp/websso/>.
[8] Jill Scharr ."Facebook, Google Users Threatened by New Security Flaw" 2 May 2014. 11 Feb 2015 <http://www.tomsguide.com/us/facebook-google-covert-redirectflaw,news-18726.html>
[9] 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" Indiana University Bloomington Bloomington, IN, USA.
[10] Guilin Wang, Jiangshan Yu, and Qi Xie (2013) "Security Analysis of A Single Sign-On Mechanism for Distributed Computer Networks" Center for Computer and Information Security Research, School of Computer Science and Software Engineering, Australia.