International Journal of Engineering Inventions
ISSN: 2278-7461, www.ijeijournal.com
Volume 1, Issue 7 (October2012) PP: 01-05

ANALYSIS OF SINGLE SIGN ON WEB – EMERGENCE OF GOOGLE SERVICE PROVIDER

Pratiba Reddy G, Ramya Mendu, Prashanth, Rama Krishna

Abstract:—Web user Applications are used widely in several fields for quality security reliability purpose the users are required toidentifyuser id and password to logging. We use a global identifier user name and password in many systems is difficult. So in different approaches are proposed to implement the problem; amongthose single sign-on (SSO) is the most popular technique. Usingthis, client can log in only once to get access to all other serverswithout log in once again. We analysed to use a single sign-on assistant calledSSOA for web application is an authentication server broker. Ifthe uservisit the web browser system using the internet explorer, SSOA validates the user id and password. SSOAHTTP POST data; HTTP header used for login,reference address and authorization URI, and then constructsHTTP POST compatible data used for validation. We also given the clear picture example for Google service provider validation by the SAML and SSOA the user can use the other applicationsand resources registered in SSOA. With which we would solve to uniquely identity authentication attaining simplicity, reliable and relatively no risk and low cost.

Keywords:—Single Sign On, Authentication, Google Service Provider, Validation.

I. INTRODUCTION

Most of the organizations started a central authentication source for internal applications and web-based portals, the single source of authentication configured properly provides strong security in the sense that users no longer keep username and passwords for different systems on sticky notes on monitors or under their keyboards. As more web services are being hosted by external service providers, the problem has reoccurred for these outside applications.[1] Users are now forced to remember username and password for HR benefits, travel agencies, expense processing, etc or programmers must develop custom code for site. Management of users becomes a complex problem for the help desk and custom built code for each external service provider can become difficult to administer and maintain.There are problems for the external service provider as well every user in an organization will need to be set up for the service providers application causing a duplicate set of data. Instead if the organization can control this user data, it would save the service provider time by not needing to set and terminate user access on a daily basis. Furthermore, one central source would allow the data to be more accurate and up-to-date. In the client/server application refer to a model for computer networking that utilizes client and server devices each designed for specific purpose can be used on the internet as well as local area networks e.g of client/server systems on the internet include web browser and web servers, FTP clients and servers, DNS. Client PCs with network software applications installed that request and receive information over the network. Mobile devices as well as desktop computers can function as clients. A server device typically stores files and databases including more complex applications like web sites. Sever often feature higher powered central processors more memory and larger disk drives than clients[2]. A client will be given access to use the resources available at the different servers only when there is a connection establishment between client and server. For connection establishment client provide the password and server verifies it. Hence there is a need for security in providing logon to the clients.

II. SECTION

2. Related Work:

The Consortium for defining standard security is Advancement of Structured Information Standards are a non-profit international organization that promotes the development and adoption of open standards for security and web services. User operating system environments evolved from traditional desktops to terminal services to centralized processing popularly achieved with single sign on solutions. Single sign on products provide apparent benefits to end users like ease of password, elimination of dependencies on desk to rest passwords and more on the other end security factor is missing ins benefits which indicates that single sign on do present security risks to the organization for example security will breach if username and password are standardized across all servers and applications. An anomaly attack to penetrate into servers can turn successful to take away information if the attacked servers have in same authentication credentials. Now the users demand interoperability outside of the enterprises own domain with outsourced services including business process, software as a service providers and with subsidiaries. Candidates today need to access different applications over the internet in execution of their daily jobs often with different usernames and passwords for everyone. It is intrinsically insecure being especially susceptible to publish worms and other malware that can quickly spread and spoil the organization.

The sign on is in place but of no use to user. Reduced sign on approach not only hurts the user, rather the meaning of SSO itself dies away. Reduced sign on approach is a byproduct of security lack in single sign on solutions.Reducing user’s complexity problems and rectifying thedecrease in adoption trend requires a balance between usersatisfaction and security. If the scale swings too far towardsecurity when trying to prevent a breach, user satisfactiondecreases. Similarly, if the scale swings toward usersatisfaction, you have to compromise IT security. In viewof decrease in single sign on adoption, Gartner offeredsome blunt advice in February 2007 that all organizationsshould look to use stronger authentication in high-risksituations such as remote access. What meant bychoosing stronger authentication not only confined thescope to setting difficult passwords but something beyondthis trivial approach.

III. SECTION

3. Problem Definition:

In general portals are become more sophisticated with complex technical and functional requirements, the issue of integrating them with legacy data sources, problems occur as an issue such as authentication. To avoid this portal need to authenticate users to back end data sources and applications may have different underlying security infrastructures a single sign –on one is efficient authentication.

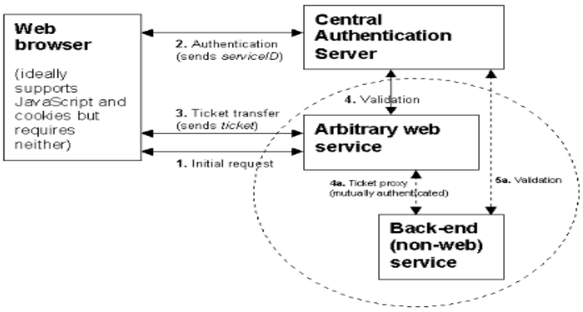
Web browser
(ideally supports JavaScript and cookies but requires neither)
2. Authentication (sends serviceID)
Central Authentication Server
4. Validation
3. Ticket transfer (sends ticket)
Arbitrary web service
1. Initial request
4a. Ticket proxy (mutually authenticated)
5a. Validation
Back-end (non-web) service

Figure 1 represents Authentication for Single Sign on

3.1. Proposed Analysis:

In general, a coherent authentication strategy or a solid authentication framework is missing. Over time this leads to a proliferation of applications, each of which comes with their own authentication needs and user repositories. At one time or another, everyone needs to remember multiple usernames and passwords to access different applications on a network. This poses a huge cost for the administration and support departments -- accounts must be set up in each application for each employee, users forget their passwords, and so on.Authentication is a horizontal requirement across multiple applications, platforms, and infrastructures. In general, there's no reason why user Mary should need multiple usernames. Ideally she should only need to identify herself once and then be provided with access to all authorized network resources.

The objective of SSO is to allow users access to all applications from one logon. It provides a unified mechanism to manage the authentication of users and implement business rules determining user access to applications and data.

Benefits of Single Sign-on

Improved user productivity: Users are no longer bogged down by multiple logins and they are not required to remember multiple IDs and passwords. Also, support personnel answer fewer requests to reset forgotten passwords.

Improved developer productivity: SSO provides developers with a common authentication framework. In fact, if the SSO mechanism is independent, then developers don't have to worry about authentication at all. They can assume that once a request for an application is accompanied by a username, then authentication has already taken place.

Simplified administration: When applications participate in a single sign-on protocol, the administration burden of managing user accounts is simplified. The degree of simplification depends on the applications since SSO only deals with authentication. So, applications may still require user-specific attributes (such as access privileges) to be set up.

Problems with single sign-on include the following:

Difficult to retrofit: An SSO solution can be difficult, time-consuming, and expensive to retrofit to existing applications.

Unattended desktop: Implementing SSO reduces some security risks, but increases others. For example, a malicious user could gain access to a user's resources if the user walks away from his machine and leaves it logged in. Although this is a problem with security in general, it is worse with SSO because all authorized resources are compromised. At least with multiple logons, the user may only be logged into one system at the time and so only one resource is compromised.

Single point of attack: With single sign-on, a single, central authentication service is used by all applications. This is an attractive target for hackers who may decide to carry out a denial of service attack.

3.2. Design Authentication for Proposed System

Authorized SSOA client should log in to authentication broker server and verify the services. SSOA is a plug-in which can be used in Microsoft Internet Explorer and Microsoft Windows browser, as well as those developed by other vendors. Authentication broker service is a web service supplied by SSOA for users. The processing logic is shown as follows.

Step 1. SSOA clients connects SSOA server using Security Socket Layer (SSL),
Step 2. SSOA server gets account and password pair from SSOA client,
Step 3. SSOA server encrypts the account and
Password using AES algorithm,
Step 4. SSOA server compares the encoded data
with the stored account and password,
Step 5. After passing validation, we access the web applications.

After validation the user can use other systems registered in SSOA. And the user can use the credential to communicate with the server. Authentication credential of SSOA server is similar to session in web service. Normally, a server will invalidate credential automatically if the user doesn't use it to access applications or resources registered in authentication broker server during a period of escaping time, e.g. 20 minutes. Authentication credential is shared by all through systems registered in SSOA, which is essentially different from the mechanism of session. The authentication broker is maintaining the credentials. By the mechanism, the server can easily determine the role of a user. The following shows the steps to add a new item. Before inserting an item, SSOA will save POST data when accessing the URI and encrypt the data using AES algorithm. Afterwards, SSOA sends them to authentication broker server to add a new item. The processing logic is shown as follows.

Step1. SSOA client connects SSOA server using Security Socket Layer (SSL);
Step2. Server receives authentication credential and encrypted from SSOA client;
Step 3. Authentication broker server gets UserID from user Credential according to CredentialID,
Step4. Set eTime in component Credential as current time of server machine plus escaping time
predetermined by the system,
Step5. Authentication server inserts component URI Broker, including UserID, URI, pData, hData, rURI and aURI.

There is a plug-in implemented in the proposed system. If the plug-in is in on mode means we can access the multiple applications without log in again. Else if the plug in is in off mode means the user can access only single application. If he tries to access multiple applications in off mode, the server loads the log in page, not the home page. The processing logic is shown as follows.

Step1. SSOA clients connects SSOA server using
Security Socket Layer (SSL);
Step2. SSOA server gets account and password pair from SSOA client;
Step3. SSOA server encrypts the account and
Password using AES algorithm;
Step4. SSOA server compares the encoded data
With the stored account and password
Step5. After passing validation, we access the web applications,
Step6. To access multiple applications turn on the
plug-in.

The user can able to create a new account, modify password and manage existing broker URI by the authentication broker server. It supports data management done by users. Taking data security into account, all data are stored in a ciphered way, which, as a result, adds more trouble in password modification. Creating new account and modify the existing account is relatively simple. The following is processing logic of password modification.

Step1. Enter the AccountID, old password, new password and confirmed new password,
Step2. After passing validation, the server judges
Whether the new password and confirmed new password is consistent,
Step3. The server gets UserID, URI, pData, hData, rURI data from user URI Broker according to UserID;
Step4. The server deciphers the data using old password as key and then encrypted using new password as key,
Step5. The server stores the newly encrypted data in user URIBroker.

IV. SECTION

4. Single Sign On Authentication for Web Google Application:

Google offers a SAML based service provides partner companies with full control over the authentication of hosted user accounts that can access web based applications like Gmail and Google mail. In this Google acts like a service provider and provides services and start of page to identify providers and control usernames passwords and other information.

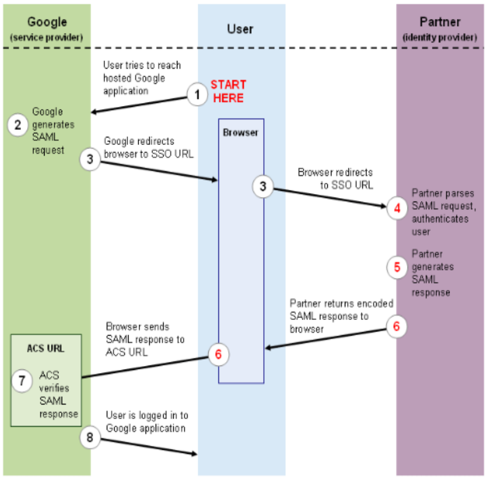
Google
(service provider)
User
Partner
(identity provider)
User tries to reach hosted Google application
1
START HERE
2
Google generates SAML request
Browser
3
Google redirects browser to SSO URL
3
Browser redirects to SSO URL
4
Partner parses SAML request, authenticates user
5
Partner generates SAML response
6
Partner returns encoded SAML response to browser
6
Browser sends SAML response to ACS URL
ACS URL
7
ACS verifies SAML response
8
User is logged in to Google application

Figure 2 shows the Authentication for Single Sign On for Web Google.

Process explains the how a user logs into host google application through a partner operated. Before this process takes place the partner must provide Google with a URL for its Single Sign On service the numbered list that follow the image explains step in more detail.
Logging into Google Apps Using SAML Transaction
The user attempts to reach a hosted Google Application such as Gmail Start Pages or Google Service.
Google generates a SAML authentication request and SAML request is encoded and embedded into URL for the partners SSO service and the Relay State parameter containing the encoded URL of the Goolge application that the user is trying to reach is also embedded in the SSO URL

This Relay State parameter is meant to be an opaque identifier that is passed back without any modification or inspection cause. Google sends a redirect to the users browser and redirect URL includes the encoded SAML authentication request that should be submitted to the partner SSO service. The partner decodes the SAML request and extracts the URL for both Google's ACS (Assertion Consumer Service) and the user's destination URL (RelayState parameter). The partner then authenticates the user. Partners could authenticate users by either asking for valid login credentials or by checking for valid session cookies. The partner generates a SAML response that contains the authenticated user's username. In accordance with the SAML 2.0 specification, this response is digitally signed with the partner's public and private DSA/RSA keys. The partner encodes the SAML response and the RelayState parameter and returns that information to the user's browser. The partner provides a mechanism so that the browser can forward that information to Google's ACS. For example, the partner could embed the SAML response and destination URL in a form and provide a button that the user can click to submit the form to Google. The partner could also include JavaScript on the page that automatically submits the form to Google. Google's ACS verifies the SAML response using the partner's public key. If the response is successfully verified, ACS redirects the user to the destination URL. The user has been redirected to the destination URL and is logged in to Google Apps.

V. CONCLUSION

To login into security is more important to protect ourdata’s from other users for those authentication mechanisms are required. System user is using different user ids andpasswords to various web applications. Use a global identifierand password in many systems is impossible to access, by using thissolution there is no need to log in to all web applications.Once we register the applications in SSOA means we canlogin in any one application and get access to all otherusers to registered web applications, eliminates the
risk of users in authentication. Here we use SSOA in websystem mainly consists of Client and Key brokerValidator Service, gateway service. Our analysis shows the example of Google service provider systems conveniently with low cost. Futurework is extended with implementation of web browser applications. Proposed system developed for use within the organization but we can extend it for WorldWide Web and also can access any application fromSanywhere without login again in reliable.

REFERENCE

1. B. Lee, H. K., and Kim, K. Strong proxy signature and its applications. In Proc. of the 2001 Symposium on Cryptography and Information Security (SCIS'01) (2001), vol. 2, pp. 603{608.
2. Bellare, M., Canetti, R., and Krawczyk, H. Keying hash functions for message authentication. In Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology (London, UK, 1996), CRYPTO '96, Springer-Verlag, pp. 1-15.
3. Bellare, M., Fischlin, M., Goldwasser, S., and Micali, S. Identi_cation protocols secure against reset attacks. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology (London, UK, 2001), EUROCRYPT '01, Springer-Verlag, pp. 495{511
4. OASIS Frequently Asked Questions “http://www.oasisopen. rg/who/faqs.php”, 2009.
5. The Twilight of Passwords: A imetable for Migrating to Stronger Authentication, Ant Allan, Gartner, Inc., Feb. 28,2007.
6. Wang, S. and Wang, H. Cyber Warfare: Steganography vs.Steganalysis, Communications of the ACM Volume 47, Number 10, pp 76-82, 2004.
7. Dodson, B., Sengupta, D., Boneh, D., and S., L. M. Secure,consumer-friendly web authentication and payments with a phone. In Proceedings of the Second International ICSTConference on Mobile Computing, Applications, and Services (MobiCASE), 2010.
8. ISO 18004:2005. Information technology (Automaticidentification and data capture techniques), QR Code 2005 barcode zymology specification Automatic.ISO, Geneva,Switzerland.
9. Single sign-on assistant an authentication broker for webapplications. Third international conference on knowledgediscovery and data mining by Fei Zhu, HongjunaDiao Schoolof computer science & Technology Soochow University.

Pratiba Reddy G pursuing Ph.D from JNTUH M.Tech computer science from JNTUH B.Tech Information Technology from JNTUH currently she is working as Head of the department IT in St.Mary's College of Engg and Technology, more than nine years of experience in Academic, she has guided many UG & PG students and conducted national technical fest. Her research areas includes Web Technologies Network Security Neural Networks.

Ramya Mendu pursuing M.Tech Software Engineering from St.Mary's College of Engg& Tech B.Tech Computer Science Engineering from Jayamukhi Insititute of Engineering & Technology. Her research areas include Networks, Data mining, Web Applications

Prashanth pursuing M.Tech Software Engineeringfrom St.Mary's College of Engg& TechB.TechInformation Technology from MaheshwaraEngg College. His research areas include Networks, Data mining, Web Applications

Rama Krishna M.Tech SoftwareEngineering from IETE New Delhi B.Tech Information Technology from RammpaEngg College Warangal. His research areas include Networks, Data mining, Web Applications