Security reliability to identify user id and password logging

[1]K. Arun Kumar,[2]N Durga Bhavani,[3]N Sujatha,[4]N Raghavender Goud,[5]P Srikanth,[6]P Karthik,
[7]P Aniket Ashok
[1]Asst. Professor, Dept. of CSE, [2,3,4,5,6,7] B. Tech., (CSE)
Malla Reddy Engineering College (Autonomous), Secunderabad, Telangana

Abstract:–

We analyzed to use a single sign-on assistant called SSOA for web application is an authentication server broker. If the user visit 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 constructs HTTP 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 applications and resources registered in SSOA. With which we would solve to uniquely identity authentication attaining simplicity, reliable and relatively no risk and low cost. Web user Applications is used widely in several fields for quality security reliability purpose the users are required to identify user 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; among those single sign-on (SSO) is the most popular technique. Using this, client can log in only once to get access to all other servers without log in once again.

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

1. 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.

SINGLE SIGN-ON

Different set of credentials (e.g., username and password) are require the user to memorize and utilize for each application the user wants to access. However this approach is inefficient and services a user has to access both inside corporation to mange potentially multiple authentication

solutions and databases individually used by each application. Further most users tend to rely on the same set of credentials for accessing all of their systems posing a serious security threat an attacker who discovers these credentials can easily assess all of the users applications.

In a single sign framework user performs unique sign-on to an identity provider trusted by the applications he wants to access. Later each time he wants access an application, it automatically verifies that user is properly authenticated by the identity provider without requiring any direct user interaction. The solution single sign-on eliminates the need for users to repeatedly prove their identities to different applications and hold different credentials for each application. Single sign-on solution significantly reduces authentication infrastructure and identity management complexity, consequently decreasing costs while increasing security

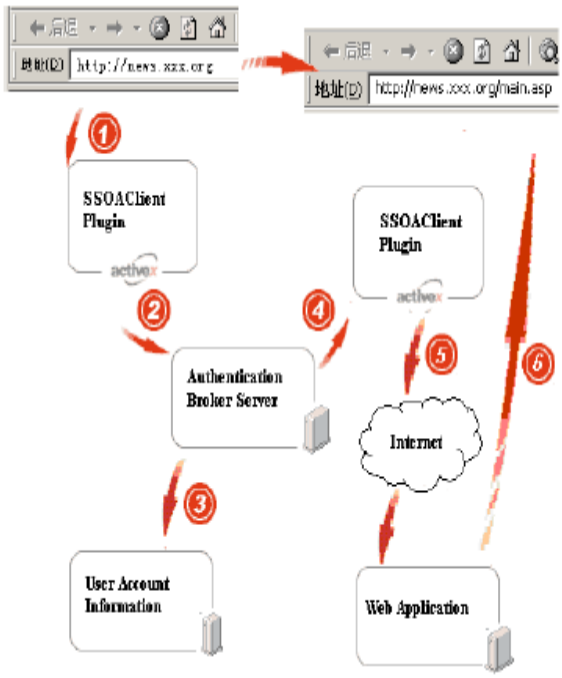

Fig. 1: Overview of authentication

We can save HTTP POST data when a user logs in a system. When the user visits the page again, the request will be intercepted by the system, and then compose HTTP POST data after necessary processing. The data after that are sent to authorization components. So that user name and password can be omitted. The login procedure can be executed by explorer monitoring program rather than the user. Fig. 1 shows when a user visits a web system using Internet explorer, the system URI is send to authentication broker server after it is captured by a monitoring plug-in installed in Internet explorer client. In step 3, authentication broker server distills HTTP POST data, HTTP header used for login, reference address and authorization URI. In step 4 and step 5, plug-in of monitoring IE in client establishes data returned by authentication broker server and constructs HTTP POST compatible data which will be sent to authorization component for users to log in. After that, the data are sent to authorization component to conduct validation. In step 6, a validated page is sent to the user.

Web Applications for Single Sign-on

One of most successful single sign-on is OpenID [4] which provides a framework for deploying flexible centralized user authentication for web applications. In OpenID user provides variety of identity which may be any website or web-based application where user already has an user account (e.g. Google). In order to sign on to a given web based application that supports OpenID user first signs on the identity provider of his/her choice and OpenID exchanges the necessary authentication data between the identity provider and application. However, it may be possible to compromise a given identity provider or the session state maintenance mechanism using simple social engineering techniques and client side or network based attacks. In order to transfer authentication

information from the identity provider to relying applications, Open ID relies on a complex mechanism involving authentication information stored as cookies in the user's machine and background HTTP requests. This mechanism can be attacked through network based techniques (such as DNS poisoning) and methods based on client side website vulnerabilities (such as cross site scripting).

Web Applications Secure Login

The secure single sign on in is snap2pass allows users to sign on to different web-bsed or networked services using their mobile phones as credentials [8]. In this framework, the users first share a secret key with a service provider, storing this key in a mobile phone running a sign on application. Each time the user wishes to log in to the service provider, he issues an authentication request and receives a random challenge encoded as a QR-Code [10]. The user then launches the log in application and acquires the QR-Code with the mobile phone's camera. The application generates an HMAC [2] of the random challenge under the user's shared secret key and sends it back to the service provider through the internet (using 3G networks or Wi-Fi). The service provider accepts the user's sign on if the HMAC is valid and use a public-key based approach where a digital signature scheme is trusted service (e.g., his internet banking website) but instead handling the user authentication challenges from other services used instead of the HMAC. Although this approach seems secure, the protocols proposed in [8] require the mobile phone to have direct online communication links to the identity provider, which increases

costs and may severely affect system performance. Also, it may still be attacked by an adversary that controls the user computer or internet connection.

2. PROBLEM DEFINITION

Nowadays web applications play vital role in the society, whenever we want to access the data in web-based need to login or sign on e.g., user form, items list, and payment details. Here we propose a solution for this problem is single sign-on with multiple log-on.

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:

Step 1. SSOA client connects SSOA server using Security Socket Layer (SSL);

Step 2. Server receives authentication credential and encrypted from SSOA client;

Step 3. Authentication broker server gets UserID from user Credential according to CredentialID,

Step 4. Set eTime in component Credential as current time of server machine plus escaping time predetermined by the system,

Step 5. 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.

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,

Step 6. 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.

Step 1. Enter the AccountID, old password, new password and confirmed new password,

Step 2. After passing validation, the server judges whether the new password and confirmed new password is consistent,

Step 3. The server gets UserID, URI, pData, hData, rURI data from user URI Broker according to UserID;

Step 4. The server deciphers the data using old password as key and then encrypted using new password as key,

Step 5. The server stores the newly encrypted data in user URIBroker.

3. SECTION

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.

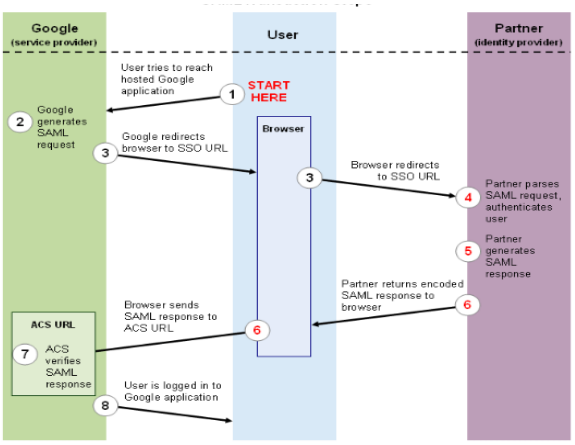

Figure 2 shows the Authentication for Single Sign On for Web Google

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.

4. 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.

REFERENCES

[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. InProceedings 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] OpenID. www.openid.net.

[5] OASIS Frequently Asked Questions "http://www.oasisopen. org/who/faqs.php", 2009.

[6] Katzenbeisser, S. and Petitcolas F.A.P. Information hiding techniques for steganography and digital watermarking. Artech House, Norwood, MA 02062, USA, 1999.

[7] Wang, S. and Wang, H. Cyber Warfare: Steganography vs. Steganalysis, Communications of the ACM Volume 47, Number 10, pp 76-82, 2004.

[8] 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 ICST

Conference on Mobile Computing, Applications, and Services (MobiCASE), 2010.

[9] ISO 18004:2005. Information technology (Automatic identification and data capture techniques), QR Code 2005 bar code zymology specification Automatic. ISO, Geneva,Switzerland.

[10] Single sign-on assistant an authentication broker for web applications. Third international conference on knowledge discovery and data mining by Fei Zhu, Hongjuna Diao School of computer science & Technology Soochow University.