ISSN 2047-3338

Web-Based System– Authentication to Single Log-on to Several Applications

Prof. N. Prasanna Balaji[1], U. Sreenivasulu[2] and C. Venkateshwar Reddy[3]
[1,2,3]Department of Information Technology, Gurunanak Engineering College, India

***Abstract*– Web Application systems are widely used in various fields for security reliability purpose the users are required to use user id and password to logging. We use a global identifier user name and password in several systems is difficult. So many 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 again. We used a single sign-on assistant called SSOA for web application is an authentication server broker. If the user visit the web application system using the explorer or Google chrome, SSOA validate the user id and password. SSOA distills HTTP POST data; HTTP header used for login, reference address and authorization URI, and then constructs HTTP POST compatible data used for validation. After the validation by the SSOA the user can use the other applications and resources registered in SSOA. With which we would solve uniform identity authentication among heterogeneous systems attaining simplicity, reliable and relatively no risk and low cost.**

***Index Terms*– Web Applications, Security, SSOA and Authentication**

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. In [3] other words client authentication is an essential criterion

II. 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

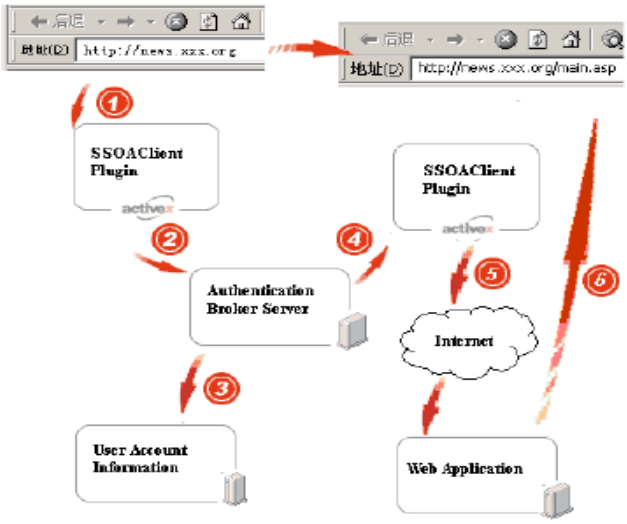

Fig. 1: Overview of authentication login the idea of Fig. 10.

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.

A. 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).

B. 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.

III. SECURITY ISSUES

User clients want to ensure that only authorized people can access and update personal information that they share with Servers. Similarly, Servers want to ensure that only authorized users have access to the services and content it provides. User authentication involves proving the identity of client to a server. Use the term “authentication” to refer to this problem. Server authentication, the task of authenticating the server to the client,

A. Deploy Ability of Secure Login

User authentication protocols differ from traditional authentication protocols in part because of the limited interface offered by the client. It is to develop an authentication system by using the protocols and technologies commonly available in today’s client and servers. The client generally speaks to the server using the Hypertext Transfer Protocol (HTTP). This may be spoken over any transport mechanism but is typically either TCP. Since HTTP is a stateless, session-less protocol, the client must provide an authentication token or authenticator with each request. Computation allows the browser to transform inputs before sending them to the server. This computation may be in a strictly defined manner, such as in HTTP Digest authentication or it may be much more flexible.

B. Performance of Secure Login

Stronger security protocols generally cost more in performance. Service providers naturally want to respond to

as many requests as possible. Cryptographic solutions will usually degrade server performance. Authentication should not needlessly consume valuable server resources such as memory and clock cycles. With current technology, SSL becomes unattractive because of the computational cost of its initial handshaking.

C. Security Requirements for Server

Server's authentication system depends on the strength and granularity of authentication desired. Granularity refers to the fact that some servers identify individual users throughout a session, while others identify users only during the first request. A fine grained system is useful if specific authorization or accountability of a user is required. A coarse-grained system may be preferred in situations where partial user anonymity is desired.

IV. 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.

A. 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.

B. Comparative Study

Web-based systems are now widely used. Enterprises, faculties, institutes and organizations usually have their own systems. These systems generally require authentication when providing service. As a result, users have to remember username and passwords of each web system, which is very

inconvenient. Some have provided approaches to alleviate the trouble, such as auto-completion supported by Microsoft, with which username and password inputted by a user for a specific URI can be automatically stored. When the URI is visited again, the password can be automatically filled by system as the username is typed or selected.

The current system proposes an alternate policy that we place an intermediate third-party service to organize the login validations at the same tine permissions to the resources before redirecting to the requested resource. In this system we are going to design a Third-party broker that validates the logins from multiple application clients who want to communicate to various resources. In this system we are taking example of designing sign-on assistant for two major domains:

i). Government Organization site

ii). Banking Site

And we provide the authentication support using the Sign-on assistant to both the client of these applications at a time. The clients of these applications may maintain uniform password schemes and also heterogeneous password management. Once they pass the login information the sign-on assistant issues a key by invoking a Web Service that performs both login validation and also validates permission to access the requested page.

C. User Access Control for Data

The following commands specify access control identifiers and they are typically used to authorize and authenticate the user (command codes are shown in parentheses)

1) User name (user)

The user identification is that which is required by the server for access to its file system. This command will normally be the first command transmitted by the user after the control connections are made (some servers may require this).

2) Password (pass)

This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control. Since password information is quite sensitive, it is desirable in general to "mask" it or suppress type out.

User access data is classified into different ways that are:

i). Client
ii). Key broker
iii). Authentication
iv). Validator Service
v). Gateway Service
vi). Server

The Client module is responsible for generating request for the specific resource. In this module we give out login information provided to the Key Broker. The client hit is redirected to the Key broker.

The Key broker module is responsible for issuing the key given by Validator Service.

Initially the Key broker just forward the login information provided by the client to Authenticator then it reaches validator and the ticket given by that service is given to the key broker. Now the request with given key is sent to Gateway service and user can access the requested resource.

The Authenticator module is responsible for validating the login and forward to the validator service. It gathers the login information provided by the Key broker via web service and performs validation and if the validation is succeeded then forwards this to validator vervice.

The validator service is a web service that provides permission check and issues a ticket to the client to access the requested resources. It checks the permission defined to access the requested resources and then issues ticket to the Key broker.

The Gateway service is responsible to route the request to the corresponding server that has the requested resource. Once Key broker gives ticket to the client then client attempts to access resource via Gateway. Then Gateway verifies ticket and forwards to the server to access the requested resource.

The Server module is the central repository that contains the resources requested by users from heterogeneous operating environments. It issues access to the request resources from the clients based on information given by the Gateway.

V. CONCLUSIONS

In security purpose is more and more important to protect our data's from other users. For that authentication mechanisms are required. Clients are using different user ids and passwords to various web applications. Use a global identifier and password in many systems is impossible. By using this solution there is no need to log in to all web applications. Once we register the applications in SSOA means we can login in any one application and get access to all other registered web applications. It eliminates the complexity and risk of users in authentication. Here we use SSOA in web-based system which mainly consists of Client and Key broker Validator Service, gateway service. The system integrates existing web applications systems. By using this, user can use heterogeneous systems conveniently with low cost. Future work is extended with web based applications, we can use many web based applications. Proposed system developed for use within the organization. But we can extend it for World Wide Web use. Then we can access any application from anywhere without login again and again in securable way.

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. 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] 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.

Prof. N. Prasanna Balaji, and Head IT has done his B.E in Computer science from Bharathidasan University, completed his M. Tech in IT (Part Time) with distinction from Punjab University Patiala, currently pursuing Ph.D in the topic "Enterprise Resource Planning" from Kakatiya University, Warangal. He has 20+ years of teaching, training and Systems Computerization. Mr. Balaji has worked as Associate Professor in CSE dept at Vignan Institute of technology & Science. At Infosys Campus Connect (two weeks residential December 2006) Programme and was recognized as one of the Best Teacher.

At Institute of Public Enterprise (IPE) he was the ERP-Incharge for Microsoft Business Solutions-Navision, and has organized a National level conference on "e-Customer Relationship Management" and three Management Development Programmes in "Recent Trends in Information Technology", two Management Development Programmes in Enterprise Resource Planning-Navission, and one Management Development Programme in Network Security for Public Sector executives. He is the co-editor for the proceedings of National level conference on "e-Customer Relationship Management". He has published and presented papers in National level Seminars and Journals.

His areas of interest are "Enterprise Resource Planning", Relational Database Management Design, Artificial Intelligence, Operating Systems, Mobile Computing, and Customer Relationship Management. He has guided many PG level and engineering students. He is also a member on various professional societies like Life Member of Computer Society of India, Indian Society for Technical Education, and a Member of International Electrical and Electronics Engineers and All India Management Association.

His present area of research is Enterprise Resource Planning. Currently he is the HOD of the Department of Information Technology in Gurunanak Engineering College, an NBA and NAAC accredited college located in Ranga Reddy Dist., Ibrahimpatnam, Hyderabad. (Email: gneccsebalaji@gmail.com)

U. Sreenivasulu He holds B. Tech., degree in Computer Science & Engineering from JNTU, Hyderabad. He obtained Post Graduation in Computer Science & Engineering in the year 2005 from SRM University, Chennai. Currently he is Assistant Professor in the Department of Information Technology in Gurunanak Engineering College, an NBA and NAAC accredited college located in Ranga Reddy Dist., Ibrahimpatnam, Hyderabad. (Email: ulsa536@gmail.com)

C. Venkateshwar Reddy pursuing M.Tech Information Technology at Gurunank Engineering College. His areas of interest include Networking, Web Application, Information Security, currently focusing on Data Mining. (Email: venkat.c.reddy@gmail.com)