ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Single Sign On Certificate Based Authentication for WS-Security

Miss. Prachi M. Kharat*, Miss. Prachi A. Deshpande and Prof. Aaditya P. Bakshi
B.E-I.T (Final Year), Jawaharlal Darda Institute of Engineering & Technology,
Yavatmal (MS) INDIA
prachikharat28391@gmail.com*, coolprachi1990@rediffmail.com and aaditya.bakshi009@yahoo.co.in

Abstract: Web services, a set of heterogeneous different technologies diverse environment used in accessing resources of server is made secure using WS-security building block used the conjunction with other web services providing requirement that describe the method to encode binary security token to accommodate variety of authentication mechanisms. In this project we are proposing Single Sing On, a secure flexible architecture, a unified authentication mechanisms for web service security needs by single point of authentication SSO is most popular schema where user login once and get accesses to all the systems to which server is connected via SSOA a plug-in is installed at client side filtering out HTTP post and header and creating a validating data returned by the authentication server. Web services security issues have become more and more important. Web services based on extensible Mark-up Languages' (XML) and related open standards, and deployed in service Oriwnted Architectures(SOA) allow a data and application implementing SSO is proposed making application at both client and server side.

Keywords: Single Sing On, Authentication , Open source**.**

I. INTRODUCTION

A single sign-on assistant for web based applications, called SSOA, aiming at 'logging in once, running everywhere', with which we would solve uniform identity authentication among heterogeneous systems attaining simplicity, scalability and relatively low cost.

This is an improved scheme[3] to the centralized approach used for integrating different authenticating schemes for achieving unified authentication. Single sign-on (SSO) is a property of access control of multiple, related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them.

When name/value pair is received, a server deals with validation according to predefined processing logic. The basic idea of the single sign-on security architecture is to shift the complexity of the security architecture to the so-called SSO service and thus release other parts of the system from certain security obligations. In the SSO architecture[1], all security algorithms[2] are found in the single SSO server which acts as the single and only authentication point for a defined domain. The SSO server, which can itself be a Web service, acts as the wrapper around the existing security infrastructure that exports various security features like authentication and authorization.

The SSO server enhances security of the whole system as the security credentials don't need to be passed around. As different applications and resources support different authentication mechanisms, single sign-on has to internally translate to and store different credentials compared to what is used for initial authentication.

SSO New concept is introduced by IEEE engineers to sign in just only once and then open all web sites without login in again and again. SSO is a property of access control of multiple, related, but independent software systems, Also the another main purpose of the single sign on facility is to provide better security.

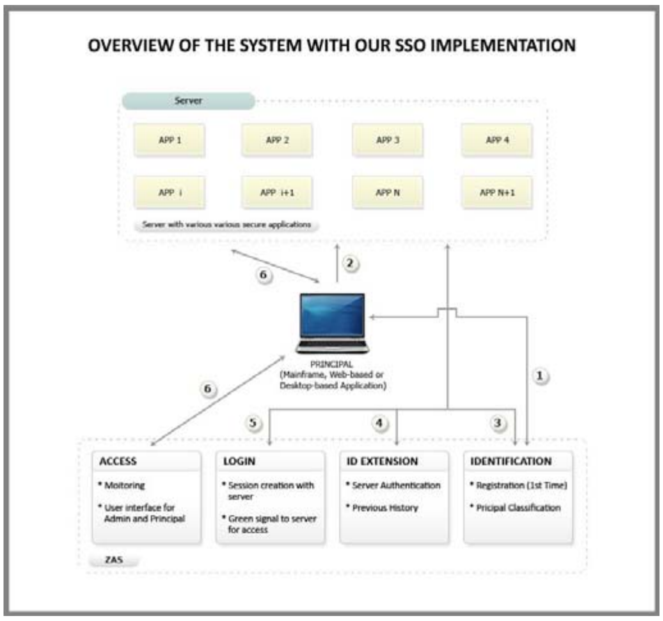
OVERVIEW OF THE SYSTEM WITH OUR SSO IMPLEMENTATION
Server
APP 1
APP 2
APP 3
APP 4
APP i
APP i+1
APP N
APP N+1
Server with various various secure applications
6
2
PRINCIPAL (Mainframe, Web-based or Desktop-based Application)
1
6
5
4
3
ACCESS
• Moitoring
• User interface for Admin and Principal
LOGIN
• Session creation with server
• Green signal to server for access
ID EXTENSION
• Server Authentication
• Previous History
IDENTIFICATION
• Registration (1st Time)
• Pricipal Classification
ZAS

Figure 1. Overview of the system with our SSO implementation.

A. *Benefits:*

SSO reduces phishing success, because users are not trained to enter password everywhere without thinking. It also reduces password fatigue from different username and password combinations and the time spent re-entering passwords for the same identity. SSO can support conventional authentication such as Windows credentials (i.e. username/password). SSO can reduce IT costs due to lower number of IT help desk calls about passwords. It also provides security on all levels of entry/exit/access to systems without the inconvenience of re-prompting users and allows centralized reporting for compliance adherence.

B. Integration:

CAS client integration components are available for all popular Web development frameworks and many popular Web applications.

C. Popular CAS Clients:

a. Java
b. Microsoft .NET Framework
c. PHP
d. Outlook Web Access
e. Confluence

The combination of open protocols and open source facilitate the development of integration components for almost any product as has been demonstrated over many years by the development of components for frameworks as varied as PL/SQL and Cold Fusion.

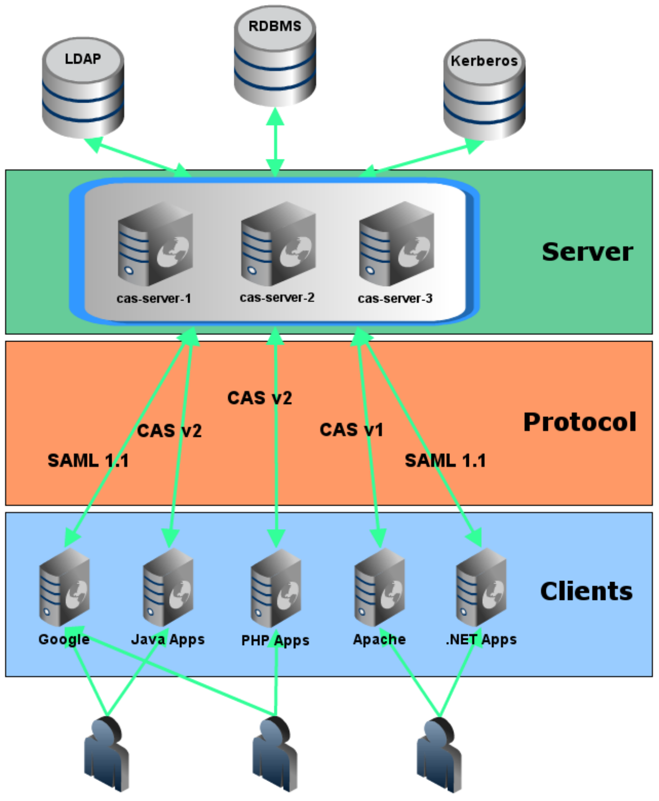

Figure 2. Web Browsing with SSO.

Web single sign-on (SSO) product composed of a single logical server component

The CAS server delegates authentication decisions to any number of supported authentication. mechanisms including LDAP/Active Directory, Kerberos, and RDBMS. The hallmark of CAS isease of integration and extension in support of a wide variety of environments. In addition to supporting a large number of technologies out of the box, the well-documented API extension points have enabled deployers to develop custom components to support novel use cases not supported by default.

II. LITERATURE REVIEW

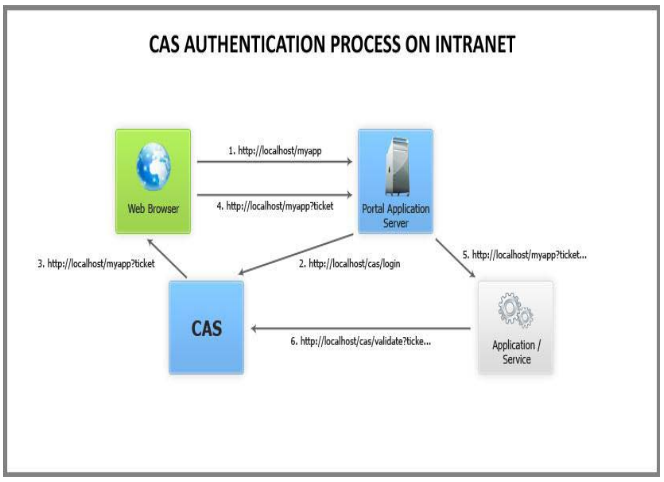
CAS AUTHENTICATION PROCESS ON INTRANET
Web Browser
1. http://localhost/myapp
4. http://localhost/myapp?ticket
Portal Application Server
3. http://localhost/myapp?ticket
2. http://localhost/cas/login
5. http://localhost/myapp?ticket...
CAS
6. http://localhost/cas/validate?ticke...
Application / Service

Figure 3. CAS Authentication Process.

SSO is used for federation access control model for each web service to get information about user whether trusted or not.SSO with trusted parties like Authentication providers some while get attacked by man in middle attacks now prevented by implementing it on trusted platform using hardware module[4]. Various SSO solutions have been proposed that depend upon PKI, Kerberos, or password-store, but they require client side infrastructure and new administrative steps.

As legacy applications have grown over the years in almost all of mid to large size organizations to support various business processes, these applications have expanded the number of users, groups and roles that need to be managed by the administrators[6].

With these growing applications, few internal users (administrators) manage most of the administration functions. This administrative burden grows as number of systems increase. This is where single sign-on (SSO) and identity management shines. Implementing SSO and identity management can reduce overall administrative work by managing user information such as their passwords, attributes, groups, and roles' memberships in a centralized user repository.

Since, HTTP is “Stateless” protocol, it is difficult to differentiate between visits to a website, unless the server can somehow”mark” a visitor. This “marking” is done by sending a piece of state information (called a cookies) to the client. Any future HTTP request made by client to the server includes this state. Many web applications have started to use cookies for session management and SSO.

A. Support multiple authentication system such as name/passwords, certificatess, one time passwords, etc[5].
B. Support web access via middle tiers to legacy applications and provide name/password mapping.
C. Scale to hundreds of thousands of users.
D. Simple to administer and configure
E. Support a whole range of deployment scenarios.

F. Only one, to a few, physical web servers, only one logical web server (multiple physical) for high availability and load balancing.
G. Many web server supporting many application but within the same domain[5].
H. Comparison with the exisiing and proposed system

Table No.1 Hence the proposed system have more features than the existing system.

Features	Existing system	Proposed system
Automatic completion	Yes	No
Access more than two applications	No	Yes
Securable	No	Yes
Multiple users	No	Yes
Session expired time	No	Yes

III. PROPOSED WORK

To provide De-centralized authentication system for all major social paltforms by using single sign on facility.

To provide better security on the web services.Provide central authentication facility.

OpenID service provider give sevice of central authentication.

Using single sign on application, we can use certificate based authentications for different clients accessing web services. A certificate is a "digital key" installed on a computer. When the computer tries to access a server, the key will be automatically presented to authenticate the user. Client certificates can be mapped to Windows accounts in either a Domain or Active Directory.

Certificate authority or certification authority (CA) is an entity that issues digital certificates.

This allows others (relying parties) to rely upon signatures or assertions made by the private key that corresponds to the public key that is certified. After validation the user can use other systems registered in SSOA.

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. Authentication credential of SSOA server is similar to session in web service.

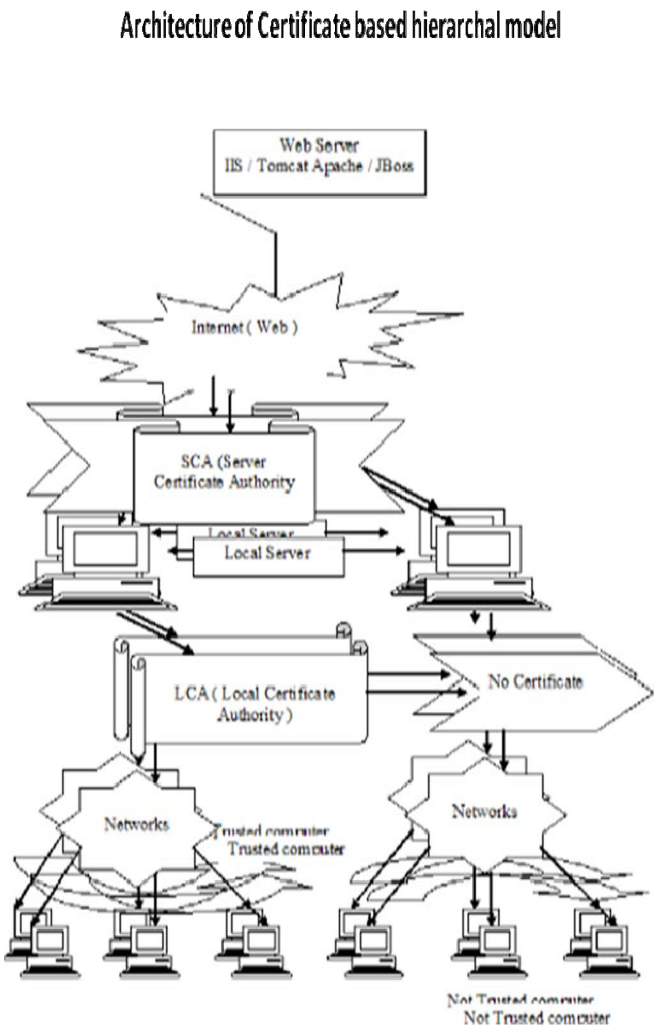
Architecture of Certificate based hierarchal model
Web Server
IIS / Tomcat Apache / JBoss
Internet (Web)
SCA (Server Certificate Authority
Local Server
LCA (Local Certificate Authority)
No Certificate
Networks
Networks
Trusted computer
Not Trusted computer

Figure 4. Architecture of Certificate Based Hierarchal Model.

A. *Advantages:*

a. *Time Efficiency:-*

Through our SSO, the authorized Thinkfinity officials are able to manage their time more efficiently. An administrator of a large, busy and popular website, who usually has a lot on his plate during any common working day, would have experienced a lot of time wastage and frustration while trying to clear his login credentials at every step. With our SSO, he can perform his duties smoothly and efficiently.

b. *Easy Management:*

Our SSO implementation made the user management very simple for the CAS. administrators. Because users only need one password, network administrators do not need to keep a list of which users have access to specific files. Additionally, administrators can set authorization levels for users so that even if they are logged onto the system with the initial SSO password, they still do not have access to every file. With these authorization levels in place, users cannot even try to guess a password to see an unauthorized file.

c. Reduced Costs:

CAS has a large user-base. Our SSO implementation saved a lot of resources that Thinkfinity would have to spend on technical support for users forgetting their different passwords for different applications. Additionally, it saved resources on security costs as well because it tracks all individuals who initially log in to the system.

Such a system can be very advantageous for an organization that runs its operations on a variety of systems from legacy enterprise architectures to new-age web-based systems. Our SSO solution can make everyday routine and working of that organization's officials very simple and easy where every authorized employee through any assigned platform in that organization can connect to the common server and database with singular set of protocols and can perform his duties without the need to sign-in repeatedly to different applications.

IV. CONCLUSION

Securing Web services is complex and possibly overwhelming. Security must be incorporated into the planned requirements from the very beginning.

Addressing a breach in security could be more expensive than implementing security measures in advance (cost of liability, public relations, loss of business, and so on). Also,security should be enforced throughout the infrastructure, both electronically and physically. Web services providers must assure their customers that the integrity, confidentiality, and availability of information they collect, maintain, use, or transmit is protected.

The confidentiality of information is threatened not only by the risk of improper access to stored information, but also by the risk of interception during transmission.

V. ACKNOWLEDGEMENT

I would like to avail this opportunity to express my deep sense of graduate and whole hearted thanks to my guide **Prof. A. P. Bakshi** for giving his valuable guidance, inspiration and affectionate encouragement to study this Paper. I also acknowledge my overwhelming gratitude and immense respect to H.O.D. of I.T. Dept. **Dr. R. M. Tugnayat,** Principal **Dr. A. W. Kolhatkar** and professors who inspired us and encouraged us whole heartedly.

VI. REFERENCE

[1]. Fumiko Satoh, Takayuki Itoh, "Single Sign On Architecture with Dynamic Tokens," 2004.

[2]. Impostor: A Single Sign-On System for Use from Untrusted Devices., Andreas Pashalidis and Chris J. Mitchell, IEEE Communications.

[3]. Cheng Yang, Jianbo Liu, Jiayin Tian, Yichun Zhang," Authentication Scheme and Simplified CAS in Mobile Multimedia Broadcast," 2009.

[4]. J.S. Park, R. Sandhu, "Binding identities and attributes using Digitally signed certificates,".

[5]. Jeremy Goold, Mark Clement, "Improving Routing Security Usinga Decentralized Public Key Distribution Algorithm," .

[6]. Wenjun Zhang"Integrated Security Framework for Secure Web Services".