Secured Identity Management using Single Sign-On in Private Cloud

Prakash V[1]Sridevi J[2]

[1]Assistant Professor, Department of Computer Science, Charan's Degree College, Bangalore, India
[2]Assistant Professor, Department of Computer Science, St. Josephs Evening College, Bangalore, India

Article Info

Article history:

Keywords:

Cloud computing
Security
Identity and Access Management
Authentication
Single Sign-On
JOSSO

ABSTRACT

There is widespread of computer networks in a distributed way and which has popularized for accessing the network services which is provided by the service providers. It is not very easy to handle different pairs of identity and related credentials, for each and every service provider, this issue will cause the communication cost in the system because both the user workload as well as the service providers' workload will increase. In order to handle this issue, it requires SSO, which is only a single login credential for multiple service providers, which would reduce the workload. This paper introduces the concept of Single Sign-On (SSO) and creation of the Single credential for multiple applications in different servers using Java Open Single Sign-On (JOSSO). Also, this work evidently proves that Single Sign-On provides security from various attacks particularly for Cloud applications.

1. INTRODUCTION

Cloud computing has become a buzz word in the current technology, it makes it possible to save it over the network and use them whenever it is needed from the internet. For instance, we have a person who stores some data online and he wants it to access it through his mobile, then storing it over the internet and accessing it would make things easier, and if somebody else needs to access it, he needs to have a proper internet connection to access the data which is stored over the internet [1].In simple wordsit can be defined as, accessing different resources through internet which uses pay as you go model. In the past years users would run their application or program from the software which is installed on their systems or on the server in their building. But due to this emerging technology that is with the invention of cloud computing, accessing these kinds of application through the internet become much easier.

1.1 Security issues in cloud computing

Although there are various advantages with the use of Cloud computing, there are many disadvantages with the same. One of the major problemswith this technology is the security problem. A secured communication over an open network is one of the major requirements. There is a better tool called Cryptography, which provides better security. The main aim of the security is data authentication, confidentiality and integrity [2]. The following section describes various security attacks in cloud environments.

1.1.1 Different attacks in cloud computing

A. Impersonation Attack

An impersonation attack is an attack in which a sender will send a data to the receiver, but during the transmission of data the third party access the data and tries to modify some content and then sends the data to the receiver. Here actually the data received by the receiver come from a third party, but the receiver will not know that the data came from thirdparties. The data is sent by third party to the receiver in the name of the sender.

B. Credential privacy Attack

The credential privacy attack is where the unauthorized person stoles the credentials of authorized person and tries accessing the application as and pretends as an authorized user. This happens during login and authentication phase. The attacker tries to make a duplicate reply which is sent to the actual user when getting a user's request for login.

C. Replay Attack

This attack is one of the major attacks which we are facing in everyday life. When the sender and receiver are communicating then the unauthorized person attacks the networks and gains the access, then the receiver will get duplicate messages, one from the actual sender and another from the unauthorized person, but in the name of the actual sender. Now the receiver will receive two messages and the receiver will not be knowing which message was received from the actual sender.

D. Denial of Service Attack

In Denialof Service attack, the attacker or third party will disrupt or interrupt the services which are sent by the server to the sender. This will be done in the name of the sender.

1.2 Authentication using Single Sign-On (SSO)

SSO is a framework which allows the users to verify and get the access for multiple application and services by just using only one credential for signing in. and the trusted third party will verify the user credential in the users' database and then gives access for accessing multiple applications [4].

1.2.1 Authentication process without SSO

Without single sign-on, every site keeps up its very own database of user and their username and password. This is the thing that happens when you attempt to sign into an application or site. The site first verifies whether users are just being validated. If the user's credentials are authenticated, then the user gets the access to the site [5].

If the user's credentials are not verified then, it asks the user to sign in and then checks user's credentials which are available in the database.Once the user logins, then, the site sends a verification information whenever the user goes through the site to make sure that the user is validated every time.The verification information is usually sent as tokens, which is very quick to process. Figure 1 and Figure 2 presents the authentication process of web applications without Single Sign-On.

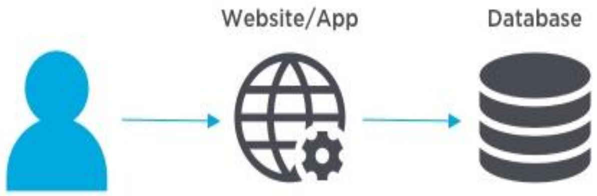


Figure 1. User requests Accesswithout SSO

The primary objective of this work is to analyze and design a secured identity management principle with SSO which is presented in section 3.1. The rest of this paper is organized as follows: Section 2 presents the related works. Section 3 presents the technical aspects and principles of SSO. Section 4 introduces the implementation of JOSSO. Section 5 discusses the results and section 6 presents the conclusion and the future extendable work of this research paper.

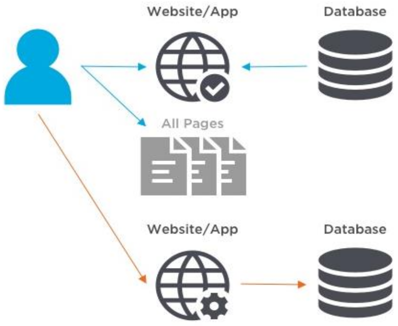


Figure 2. User is granted access and requests to access the new site with the credentials

2. RELATED WORK

This section presents different literatures which helped this work for design and implementation of SSO in various architectures. The work of [6] presents an importantAsymmetric RSA algorithm which uses hash functions, in order to find the timestampand to reduce the running cost of the system. This method is very efficient in performance and transmission cost. And, the parameters considered are performance and communication cost. So, this method is unreliable to impersonation attack.

This work [7] had demonstrated that RSA based scheme is not that secure by applying impersonation attack and credential backing attack without any credential. [8] has used key agreement and cryptosystems. One of the benefits of this methodology is, according to the user's requirement the user can choose their password, and whenever it is necessary the user can change the password. There is no synchronization problem, which is a nonce based scheme. User and Server can mutually authenticate each other. The research[9] focuses on developing mindfulness and concerns regards to Cloud Computing and Information Security, there is developing mindfulness and utilization of Security Algorithms into information frameworks and procedures. Also, shows a brief overview and comparison of Cryptographic Algorithms, with an accentuation on Symmetric algorithms which ought to be utilized for Cloud based applications and administrations that require information and connection encryption. In this paper, a survey was made on Symmetric and Asymmetric algorithms which focuses on Symmetric Algorithms for security thought on which one should be utilized for Cloud based applications and administrations that require information and connection encryption.[10] deals about Security-as-a model. The focus is on security delivered as cloud services; for example, security gave through the cloud rather than on premise security arrangements. Identity and Access Management (IAM) focuses around validation, approval, organization of Identities and review. Its essential concern is confirmation of personality or identity and granting the right degree of access to resources which are ensured in the cloud condition. The IAM executed as the cloud service can profit the client with every one of the preferences offered by Security-as-a-service (SECaaS).

3. PRINCIPLE OF SINGLE SIGN-ON MECHANISM

Single sign-on is a process where the user can use only one credential to access multiple applications. There is one trusted third party who verifies all the credentials of the user. There is no need for the user to remember multiple user name and passwords [11].

3.1 Authentication with SSO

Authentication with SSO depends on a trusted relationship between the domains. With single sign-on, this is the thing that happens when you attempt to sign in to an application or site.First and foremost, the site will validate that user have been authenticated or verified by the Single sign on, if the user is authenticated then it will give the access to the site, if not it will be sent back to the SSO sign in page in order to sign in. User enter the single username/password word that you use for access. SSO verifies your identity and sends a notification to the SSO. Figure 3 presents the overall architecture of the SSO authentication process.

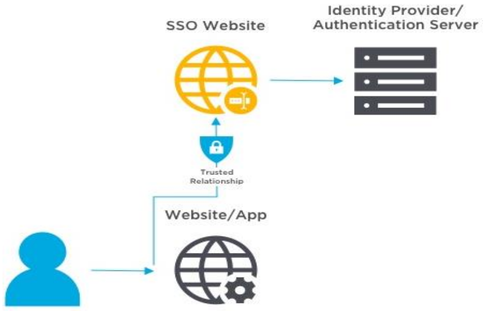


Figure 3. User requests Access for web application

The SSO passes validation information to the site and returns you to that site. In SSO, the confirmation check information appears as tokens. The site redirects the user to the SSO site to log in. The user will log in with a single username and password. Then the SSO site authenticates the user's identity with identity provider. Figure 4 represents the interaction between the authentication server and the application server [12].

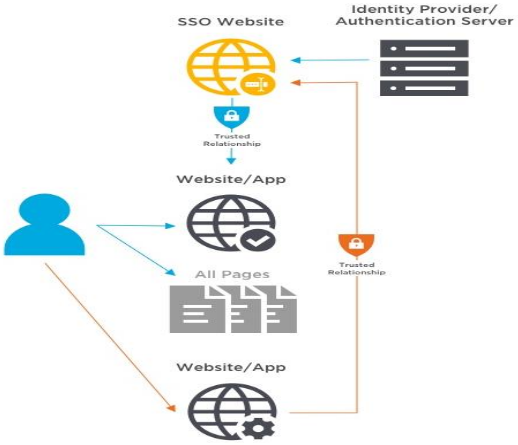


Figure 4. User is granted access by authentication server and then request to access the new site

3.2 SSO approaches

There are three main categories of SSO [13]:

1. Password Synchronizer: In this category, in order to access multiple system or application the user needs a different username but at the same time, a common password.
2. SSO enterprise: This category provides an opportunity for the user to simplify the complexity of its credentials.
3. SSO web: This category deals with SSO in web, here in order to get the relationship between many different organizations SSO web is used. It uses some standards such as windows authentication.

Figure 5 shows the different approaches for deploying SSO effectively.

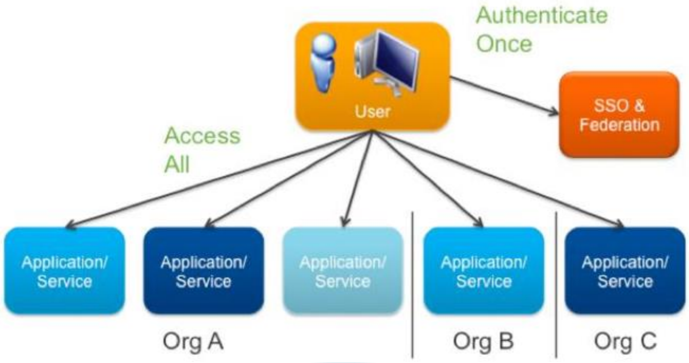


Figure 5. SSO approaches

3.3 Phases of secured authentication scheme

There are three important phases are associated with SSO implementation which is given as follows:

A. Phase 1 - *Client side*

It is the user who wanted to access the application, who will be entering the user credential. The user will need to register himself or herself to use SSO.

B. Phase 2 - *Authentication party*

Authentication party is the one which authenticates the user by checking the user identity and credentials. Once registration is successful the user can login to the system to access multiple application through SSO.

C. Phase 3 - *Server side*

This is the place where the user's credentials are stored. This phase deals with verifying the user's credentials. Also, allows the user can customize the credentials whenever is needed.

Figure 6 shows the flow of processes of these phases involved in the authentication process using JOSSO.

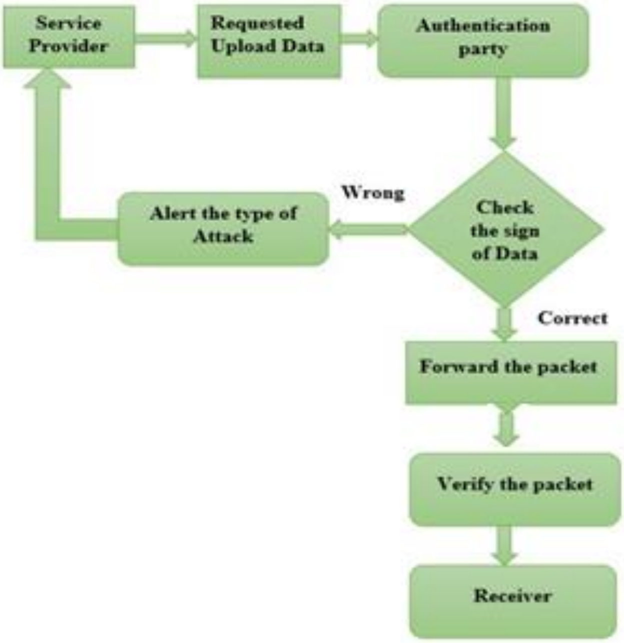


Figure 6. System flow diagram of the authentication process in JOSSO

4. IMPLEMENTATION OF SECURED IDENTITY MANAGEMENT USING JOSSO

Java Open Single Sign-On (JOSSO) is an open source Single Sign-On solution which is used for the implementation of single sign-on. JOSSO also gives permission for accessing web application as well [14].Without requiring multiple authentications by the user, the user has the permission to access multiple applications and many more services from different servers, which can be succeeded by SSO. This methodology reduces all complexities such as time and also avoids from signing again and again for different application.

4.1 Experimental setup

To deploy SSO using JOSSO, the user needs to follow the following steps:

Step1: Download JOSSO from the any internet source
Step2: Once JOSSO is downloaded, JOSSO should be installed in the system. They are two ways of installing:
Option 1: open the downloaded file**.** The file can be opened by double clicking the downloaded windows.jar file.
Option 2: Command prompt should be opened.

Go to the location where JOSSO was downloaded, check the jarfile. Then need to execute jar file.This is the location where the JOSSO installation files are stored: C:\atricore\josso-ce-2.4.3-SNAPSHOT
Use the following URL which will be shown at the end of installation: http://localhost:8081/atricore-console.

Step3: In order to create SSO, follow the steps:

i. Go to Command Prompt (Run as Administrator)

ii. Execute the following commands to browse to JOSSO location:

a. cd\
b. cd atricore
c. cd josso-ce-2.4.3-SNAPSHOT
d. cd bin
e. atricore

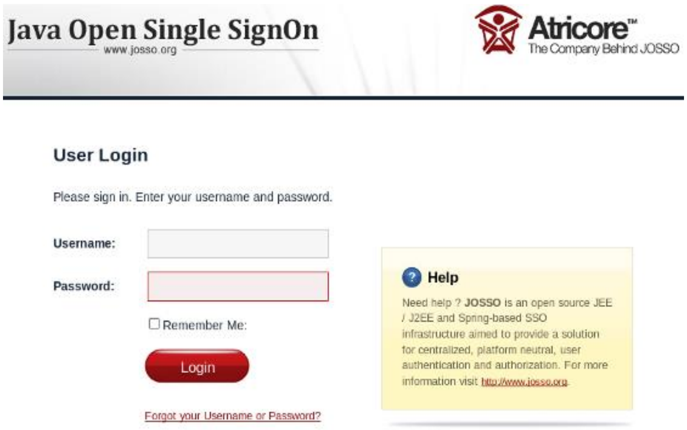
When org.josso.org (Log Listener) shown in command prompt, then it is ready to access the URL above.

iii. Open the browser.

iv. Navigate to http://localhost:8081/atricore-console.

v. User might be asked to run Flash.

vi. When the login screen pops up, give the proper credentials. Figure 6 shows the login screen of JOSSO.



Java Open Single SignOn
www.josso.org

Atricore™
The Company Behind JOSSO

User Login

Please sign in. Enter your username and password.

Username:

Password:

☐ Remember Me:

Login

Forgot your Username or Password?

Help

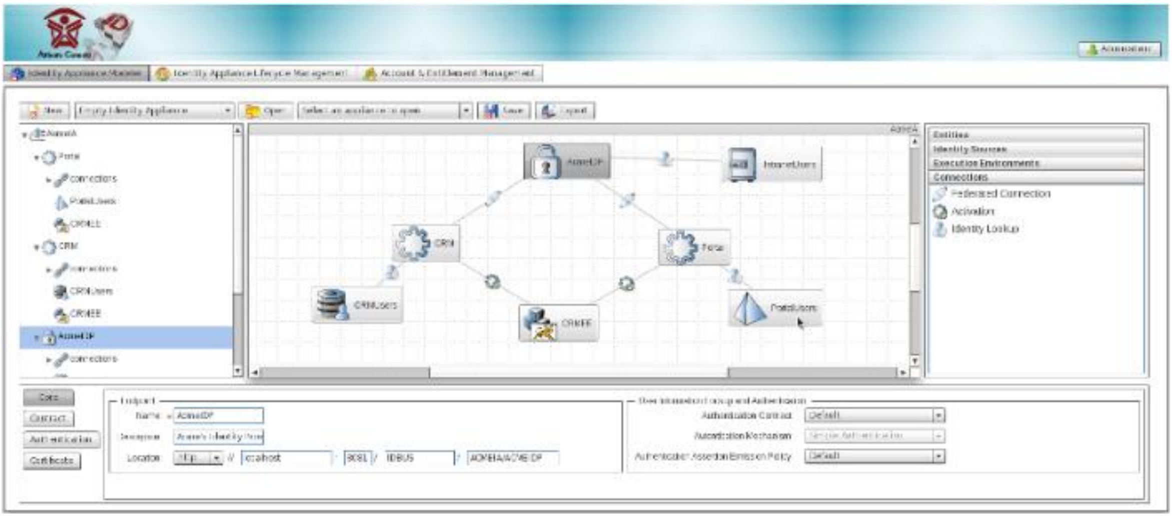
Need help ? **JOSSO** is an open source JEE / J2EE and Spring-based SSO infrastructure aimed to provide a solution for centralized, platform neutral, user authentication and authorization. For more information visit http://www.josso.org.

Figure 7. Login with JOSSO

vii. Designing Identity Appliance Modeler

In order to tweak the execution, the client required to deal with low level artifacts like XML descriptors while the JOSSO gives console to provisioning SSO support onto the ideal condition. This empowers a high section boundary. To breathe life into an undertaking the client answerable to the personality design must be subject to capable individuals which is by and large not SME specialists because of absence of perceivability or control of the combined single sign-on setting.

As a result, the pace of miscommunication increments, and this prompt noteworthy hazard in the character and access the board venture. Figure 8 presents a sample Identity Appliance Modeller with authentication server and a web application is shown.



Name: AcmeIDP
Description: Acme's Identity Prov
Location: http :// localhost : 8081 / IDBUS / ACME/A/ACME-IDP
Authentication Contract: Default
Authentication Assertion Emission Policy: Default

Figure 8. Identity Appliance modeler

4.2 Configuring web servers

In this work, we configured 2 web servers to simulate two web applications running on them separately. We used Tomcat 7 and Tomcat 8.5 versions. Install and configure the web servers as follows:

Step1: Installation of Tomcat 7

- Download Tomcat7 from internet source.
- During installation, let the default JRE be selected Choose Components: Service Start-up, native default ports are there as it is.
- Server Shutdown Port – 8005 HTTP Connector Port - 8080AJP – 8009
- Set Username & Password as admin
- Installationlocation: C:\Apache\Tomcat7
- Open the browser: http://localhost:8080 (Check whetherTomcat 7 is working)

Step2: Installation of Tomcat8.5

- Download source: https://tomcat.apache.org/download-80.cgi
- During installation, let the default JRE be selected.
- Choose Components: Service Startup, Native
- Configure Ports as follows: Server Shutdown Port – 8015 HTTP Connector Port - 8090 AJP – 8019.
- Set Username & Password as admin.
- Installation location: C:\Apache\Tomcat85 (You need to change ports because default is used by tomcat 7)
- Open the browser: http://localhost:8090 (Check whetherTomcat 8.5 is working)

Step3: Configure Tomcat 7 and Tomcat 8.5

The path C:\Apache\Tomcat7 and C:\Apache\Tomcat85 are the locations where Tomcat 7 and Tomcat 8.5 are installed respectively. You should insert your installation path as:<Your-Installation-Directory>\conf\jaas.conf

4.3 Working with JOSSO

Once the web servers are configures and Atricore is ready with modeler, follow the given steps to work with the designed secured identity management system. We used idtest as a sample web application in our research [15].

Step 1:

- Start Tomcat 7 and Tomcat 8.5 and check whether it runs via a browser:
- http://localhost:8080/ (Tomcat 7)
- http://localhost:8090/ (Tomcat 8.5)

Step 2:

- Open Atricore console in browser (after executing atricore in command prompt).
- Open http://localhost:8081/atricore-console and login with admin and atricore.

Step 3: Download idtest3.zip which is available in the atricore installation location.

Step 4: Click import and select idtest.zip.

Step 5: Open idtest if not opened.

Step 6:

- In the modeller, click tom1
- Bottom screen, in Install Home: set your Tomcat 7 path and choose Version as 7.0. Example: Install

Step 7: In the modeller, click tom2 and setInstall Home: set your Tomcat 8.5 path and choose Version as 8.5.
Example: Install Home: C:\Apache\Tomcat85

Step 8:

- In the modeller, click josso1
- Check location for a sample application partnerapp: localhost, 8080, partnerapp. (http://localhost:8080/partnerapp)

Step 9:

- In the modeller, click josso2
- Check location: localhost, 8090, partnerapp
- (http://localhost:8090/partnerapp)

Step 10: Click Save. You have to now install demo applications & reactivate tom1 & tom2 execution environments.

Step 11: Click tom1. First check overwrites the original, install the demo and then check reactivate

Step 12: Click tom2. First check the overwrite original, install the demo and then check reactivate.
Check simultaneously in command prompt to see whether demo applications getting installed. It will be available at <Tomcat-Path>/webapps/partnerapp (Check in tomcat 7 and tomcat 8.5).
The partnerapp in both locations is installed by JOSSO for the convenience as a demo.
Step 13: Go to Accounts & Entitlement Management tab and set the following:
(1) In Identity Vault, check if Default Identity Connector is selected.
(2) Under Manage Groups, click Create Group – role1 (Role1 - JOSSO community edition is set to work for this group alone).
(3) Under Manage Users, click Create Useras follows:
Under Groups, drag role1 from Available Groups to Member.
Under Passwords set the Password and then Click Save.

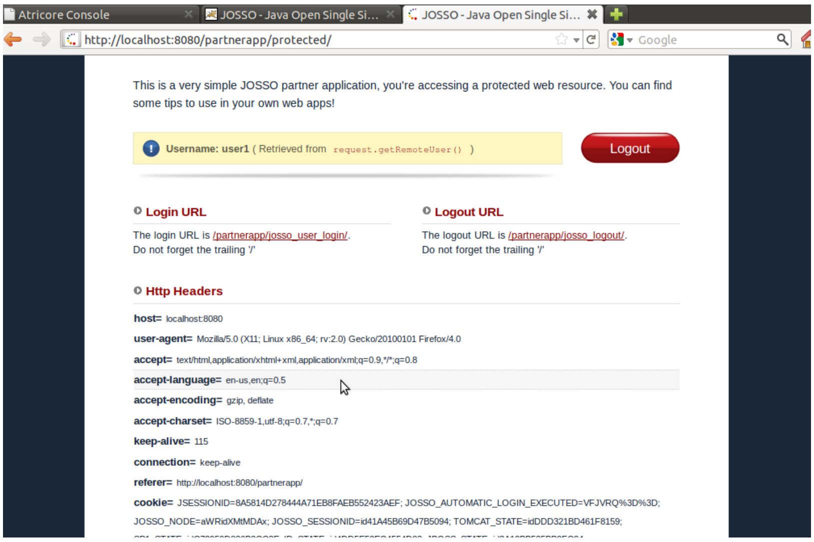
Step 14: Go to Identity Appliance Lifecycle management tab, set the following:

(1) Drag idtest3 from Saved to Staged.
(2) Drag idtest3 from Staged to Deployed.
(3) Start identity appliance.

Step 15: Access via Browser to see the appliance working:

(1) Open: http://localhost:8080/partnerappand http://localhost:8090/partnerapp
(2) You should see "anonymous" login page in both URLs
(3) Click Login and give credentials in any one application (any one URL)
(4) When you refresh the other application, you will automatically be logged in.

Figure 8 shows the login screen of a user. The same credentials will be passed to the other connected web applications to the authorization server.


Atricore Console
JOSSO - Java Open Single Si...
JOSSO - Java Open Single Si...
http://localhost:8080/partnerapp/protected/
Google
Logout

This is a very simple JOSSO partner application, you're accessing a protected web resource. You can find some tips to use in your own web apps!

Username: user1 (Retrieved from `request.getRemoteUser()`)

Login URL

The login URL is /partnerapp/josso_user_login/.
Do not forget the trailing '/'

Logout URL

The logout URL is /partnerapp/josso_logout/.
Do not forget the trailing '/'

Http Headers

```
host= localhost:8080
user-agent= Mozilla/5.0 (X11; Linux x86_64; rv:2.0) Gecko/20100101 Firefox/4.0
accept= text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-language= en-us,en;q=0.5
accept-encoding= gzip, deflate
accept-charset= ISO-8859-1,utf-8;q=0.7,*;q=0.7
keep-alive= 115
connection= keep-alive
referer= http://localhost:8080/partnerapp/
cookie= JSESSIONID=8A5814D278444A71EB8FAEB552423AEF; JOSSO_AUTOMATIC_LOGIN_EXECUTED=VFJVRQ%3D%3D;
JOSSO_NODE=aWRidXMtMDAx; JOSSO_SESSIONID=id41A45B69D47B5094; TOMCAT_STATE=idDDD321BD461F8159;
```

Figure 8 JOSSO authorization of a user

5. RESULT

By implementing all the above steps in JOSSO, there is the successful outcome of SSO, where a user's single credential is used to access multiple applications in different servers. With the help of SSO there is no need to remember multiple credentials to access different applications.

6. CONCLUSION

The main focus of this research paper is to introduce the key principles of SSO and the technique of implementation of SSO using JOSSO. The process of communication between the users and the service providers through the Internet leads to many security issues and attacks. This research work has implemented using SSO in JOSSO where a single user credential could be used to access multiple applications of the different servers. If the user gives proper credential and logs in to one application, then again and again the user need not have to enter the username and password for accessing another application in different server, using a single pair of credential multiple application can be accessed from different servers, and also SSO provides security by preventing the attacks which is discussed in this paper. It is observed that the major cloud service providers tend towards providing their services using SSO principle.Future expansion of this work is to implement Multi Factor Authentication (MFA) along with SSO to enhance the security and prevent other attacks.

REFERENCES

[1] A. Velte, T. Velte, R. Elsenpeter, Cloud Computing - A Practical Approach, McGraw-Hill Education, 2009.

[2] Abbas, H., Maennel, O., & Assar, S. "Security and privacy issues in cloud computing", Annals of Telecommunications, 72 (5-6), 233–235, 2017.

[3] Thakare, V. R., & John Singh, K. "A Study of Security and Privacy Issues at Service Models of Cloud Computing", Indian Journal of Science and Technology, 9 (38), 2016.

[4] Gunasekhar, T., Rao, K. T., &Basu, M. T., "Understanding insider attack problem and scope in cloud" In: Proc. of International Conference on Circuits, Power and Computing Technologies, 2015.

[5] Wang, G., Yu, J., &Xie, Q, "Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks" IEEE Transactions on Industrial Informatics, 9 (1), 294–302, 2013.

[6] W. Juang, S. Chen, and H. Liaw, "Robust and efficient password authentication key agreement using smart cards", IEEE Trans. Ind. Electron, 15 (6), 2551-2556, 2013.

[7] Sanket Bhat, SaumitraDamle, "Kerberos An Authentication Protocol", International Journal of Advance Research in Computer Science and Management Studies, 200-204, 2014.

[8] Bhardwaj, A., Subrahmanyam, G. V. B., Avasthi, V., & Sastry, H., "Security Algorithms for Cloud Computing", Procedia Computer Science, 85, 535–542, 2016.

[9] Deepak Sharma, Chandrashekar Dhote, "Identity and Access Management as Security-as-a-Service", International Journal Engineering Science and Technology, Volume 21, Issue 4, 2018.

[10] Jean Jacob, Mary John, "Security enhancement of single sign on mechanism for distributed computer networks", International Journal of Modern Engineering Research, volume 3, Issue 3, 1811-1814, 2013.

[11] Michel Abdalla, David Pointcheval, "Simple Password-Based Encrypted Key Exchange Protocols", in Springer Verlag, Volume 3376, 191-208, 2005.

[12] Chang, H., & Choi, E., "User Authentication in Cloud Computing", Ubiquitous Computing and Multimedia Applications, 338–342, 2011.

[13] Sabahi, F., "Virtualization-level security in cloud computing", In: Proc. of IEEE 3rd Int. Conf. Communication & Software Networks, pp. 250–254, 2011.

[14] Available Online at: http://www.josso.org/architecture.html

[15] Shruti Bawaskar, "Enhanced SSO based Multi-Factor Authentication for Web Security", International Journal of Computer Science and Information Technologies, Volume 7 (2), 960-966, 2016.