ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks

Noqaiya Ali

PG Student [CSE], Department of Computer Science, PESIT, Bangalore, India

ABSTRACT: Single Sign-On (SSO) is a mechanism that allows users to sign on or log in only once and have their identities automatically verified by each application or service which they may wish to access later on. Nowadays, the applications being used have an architecture which may not support the mechanism of Single Sign-On. These applications need the user to remember and have different set of usernames and passwords for different applications. But, this approach is both inefficient and inconvenient with the ever increasing or as one can say the exponential growth of the number of applications and services provided. SSO is an authentication mechanism that allows a user with proper credentials or in other words a valid user with a single credential to be authenticated by multiple service providers so that they would have to maintain only a single set of credentials, in a distributed computer network where there are different service providers offering various services. Thus, user authentication has a very important role in the verification of user validity. Here, a token generation method is used to generate a token and encrypt it by symmetric encryption technique. This data along with the user id and the IP address of the user's machine and the login time of that particular user is sent to the other applications for Updation in their database. IP address checking assures Single Sign-On for the user from his system. The login time determines as to how long the token would be valid so that a particular user can have continued access to his/her account from his personal system without having to enter their credentials. Thus, implementing a safe and a secure authentication protocol with these complex security properties does not come easy.

KEYWORDS: AES; authentication; confidentiality; single sign on

I.INTRODUCTION

A Single sign-on (SSO) is a mechanism which allows a user having a single credential i.e. only one set of id and password to be authenticated by various service providers in a distributed computer network. Users should have access to the services provided by the service providers. It helps in checking whether a user is both valid and authentic and whether they can be granted access to the services requested by it.

Confidentiality of the data exchanged between a user and service provider requires the implementation of safe and secure authentication protocols but with these complex security properties it is not easy. Asking a single user to have various ids and passwords for different service providers would just cause an increase in the workload of both the users and the service providers. Thereby causing communication overhead of the networks. The single sign-on (SSO) mechanism provides a solution to this problem. Thus, after having proper credentials which would be valid for a short period (say one day), the user can access services provided by different applications on different servers for that time period without logging in again and again from his personal system.

The 3 basic properties of SSO are Unforgeability of data, User credential safety and Soundness. *Unforgeability* of data means that once a user has registered themselves they have a particular id and a particular password which belongs to them, thus no one will be able to forge a valid credential for an already existing user. Registered users will only have access. *User Credential Safety* ensures that fake service providers will never be able to recover or obtain the credentials of a valid user. Thus, they will not be able to use it log onto the applications and make use of the services provided by

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

the applications running on those service providers. *Soundness* of this system is that if a user does not have an account in an application running on one server then without the valid credentials they will not be able to make use of the services provided by that application running on that service provider.

The main contribution here is the design of mechanism that aims to increase accessibility and security and to be able to minimize the overhead of having too many credentials for each application on each sever by creating a mechanism by which signing into one application on one server will result in logging into another application on another server (when opened) if the user has an account in both. Thereby providing easy accessibility to applications on different servers and maintaining the security of data being transferred from one server to another. It also reduce the communication overhead of network.

II. LITERATURE SURVEY

Single sign-on (SSO) is a described as a particular session or a user authentication method that allows a user to remember only one set of credentials that is just one id and password in order to have access to services provided by multiple applications. Single sign-on (SSO) is a property of access management of various independent software systems connected over a network. The user makes use of this property in order to log in once and gain access to various services by various applications without providing his/her credentials again and again. A simple and easy version of single sign-on is possible to be achieved by using cookies but it is only possible if the sites are present on the same domain. A Survey on Single Sign-On Mechanism for Multiple Service Authentications [1] shows that single sign-on (SSO) is a methodology that provides access control that enables a user to log in just once thereby gaining access to the resources of multiple software systems without having to log in again. Single sign-off works also in a way whereby a single action of signing out from one application terminates access to other applications also.

Different applications and resources support different authentication mechanisms. Therefore, single sign-on has to keep a check and store credentials in a manner that it can authenticate that user the next time. Users can thus make use of the various network services. If a user's credentials are valid i.e. he is a registered user then he will be able to use the services. There may be some fake servers, so to avoid them; users will have to authenticate the service providers even. After mutual authentication a session key is negotiated. This helps in keeping the data exchanged confidential.

However, it is a major task to design an efficient and a secure authentication mechanism with these security properties which would work in complex computer network environments. The current application architectures work in a different way. Here the user has to memorize and use a different set of credentials for each application they wish to use. However, this approach is inefficient and insecure and with the rapidly multiplying growth in the number of applications and services a user has to have access to each application with the help of just a single set of credentials. Another study on the security analysis of a single sign-on mechanism for usage in distributed computer networks [2] was done.

In 2000, Lee and Chang [5] proposed a user identification and key distribution scheme which would maintain user anonymity. Later, Wu and Hsu pointed out that Lee-Chang scheme is insecure against both impersonation attack and identity disclosure attacks [6]. Meanwhile, a weakness was identified in Wu-Hsu scheme [7] and an improvement was proposed for it. But these schemes were insecure under identity disclosure attack, and an RSA-based user identification scheme to overcome this was proposed and further implemented. It has been seen that user identification forms an important access control mechanism for client-server networking architectures. Some user identification schemes have been proposed for this mechanism. Even then, maximum of the current existing schemes are unable to preserve user anonymity under attack. In order to deal with these issues a secure and safe SSO mechanism was proposed. It is both efficient and secure. This would work for mobile devices in distributed computer networks [3]. Even entity authentication and also authenticated key exchange are an issue when it comes to security in distributed computer networks. Here we get an overview [4] of the first treatment of entity authentication and authenticated key exchange. The major problem dual authentication setting is the mutual authentication and key exchange among the user and service provider. Finally, here, in this paper we are showing a single sign-on as a way of using on one's personal

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

system with one set of user id and password and without having to log in again into another application. The method used is symmetric encryption or AES which provides an encryption process of securely and safely using Single Sign-On on two applications running on two servers.

III. METHODOLOGY

The system has two servers with two applications running on each of them. Namely File System Application running on Windows XP and a Mail System Application running on Ubuntu.

A user having an account in both the applications, on logging into File System can then automatically get logged into Mail System on opening the page.

The various modules involved in this mechanism are:

1. Registration.
2. Token Generation and IP Address Passing
3. Token Update
4. Auto Login

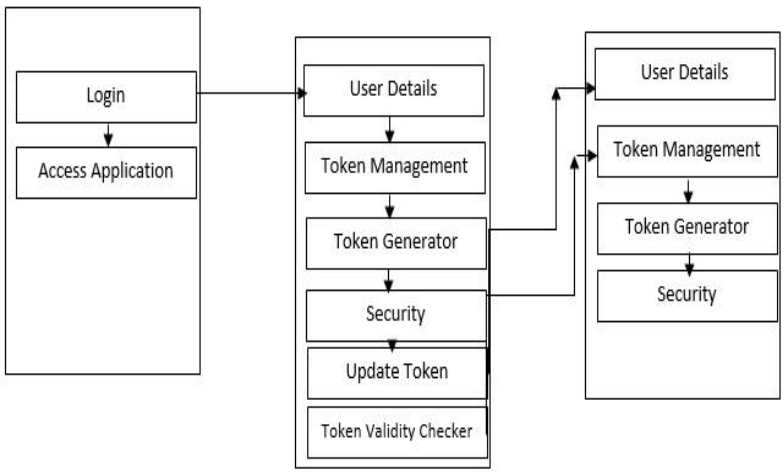

Figure 1: Architecture of the Proposed System

The detailed description is as follows:

Registration

User can register to access both the applications running on two different servers. User can register with Mail System application using his email ID. User can register with File System using the same email ID. The details of the user provided to a particular application will be saved in the corresponding server's database.

The user details might be different for different application but if the user wants to use the automated login system i.e.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Single Sign-On, then he/she must register with the same email ID. User can login using the same email ID in both the applications.

Token Generation

When the user logs in into one application, server retrieves the user machine's IP address by a method called getRemoteAddr (). Each system has a unique IP address on being connected to a network. This IP address is retrieved by the server when the system is up and running. If the system is off, even then another user can use the same email id but not the same IP address because the other user's system will have a different IP address. Thus the fake user cannot login into anyone else's account since the IP for that particular user as stored in the database is different.

The server then checks whether the IP address is present in its own database or not. If not present the server generates the token using the token generation algorithm. If the IP address is present in the database then server checks the last log in time.

If the difference between last login time and current time is less than 24 hours, then server redirects the user home page to the user. If the 24 hours session gets over, then server generates the token for the user again using the token generation algorithm.

Token Update

Once the server generates the token for a user, it encrypts the token using AES encryption. While encryption it uses the email as key and an encrypted token is generated. This encrypted token along with the email id of the user and the time of login is sent to the other server following the HTTP protocol. The next server after receiving the encrypted token decrypts it using the received user ID and verifies with registered users from the database. If the user with same email ID is already registered, then it updates the token and the IP address and its time of login. While storing the token in the database the time of validation of token is also set to 24 hrs from the login time.

Auto Login

If the user accesses the URL, then at server side, the IP address of the user is checked. The Server after fetching the IP address, search the IP address in the database. If the IP address is present in the database, server checks the user token validation. If the session is still active, server directly redirects to the user home page.

The various methods and detailed steps involved in each module are as follows:

Gathering Client Information

Step1: User sends the request to the server to a particular URL

Step2: Server retrieves the IP address of the client machine from the HttpServletRequest object using the getRemoteAddr () method

Step3: The application updates the IP address of the client machine and email ID in the status table and send the details to the other application in the another server.

Step4: In the other application if the client sends the request, the server side controller fetches the IP address from the request object using the getRemoteAddr() method and checks if the IP address and the email of the client is present in the status table or not
Step5: If the IP and email is present in the status table then server sends the home page of the client

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Token generation process

Step1: When the user logs in to server 1, token is generated and the time of validation is set as 24 hrs. Step2: One process is created in the server that deletes the token after 24 hrs.

Step3: Server 1 then sends the token to the server 2.

Setp4: After receiving the token by the server2, it verifies the user identification (Email). Step5: Server2 then checks the database for any old token.

Setp6: If the token is present in the database it updates and if not present then it inserts the token for the user. Step7: While storing the token in the database, it sets the time of validation of the token for 24 hrs.

Step8: When user tries to access the application deployed in the respective server, server checks the validity of the token.

Step9: If the token validity is not expired, server redirects the home page of the user and if it expires then it create a new token and updates in the database and sets the validity for 24 hrs.

Step10: The current server updates the new token for the user to another server.

Token Encryption process

AES or Advanced Encryption Standard is used for encryption.

The token generation mechanism is used to generate a token for a particular user on server1. That token is then stored in the status table of File System application's database. The token is then encrypted using AES. The user email id is used as a key for the encryption process. The email id is first converted to a byte array and encrypted by AES again to convert it into a key by which the token is encrypted.

After encryption of the token, it is converted into bytes. Obviously, the token will not be sent to the second server in byte format, so it is converted into a string and encoded by Base-64 encoder and then it is sent to the second server where it is decrypted with the same key and stored in its database. AES is used for encryption and decryption in Single Sign-On.

Advantages Of Using AES

• Extremely Secure: Only encryptor and decryptor have the key. It would take billions of years for the hacker to figure out the key used for encryption

• Relatively Fast: Encrypting and decrypting of symmetric key data is easy to do. Thereby it provides a good reading and writing performance.

Limitations Of AES

• Sharing the key: The most important problem which arises with symmetric key encryption is the requirement to have a safe way to transfer or communicate the key to the other party with whom the data is being shared. Encryption keys are not just simple strings of text. They are blocks of gibberish data formed after encryption. There has to be a safe way in order to transfer the key to the other party. Had a safe way existed, there was then no need of encryption, but since it

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

doesn't, thereby it causes a limitation. Thus, symmetric key encryption is better used while encrypting our own information since no key sharing is involved.

• Damage if system gets compromised: If someone gets the symmetric key then they can decrypt everything encrypted by the key. When using symmetric encryption for a two-way communication, both parties suffer loss of data, both on the sending and receiving end. But, with asymmetric public-key encryption, this problem gets minimized. This is because even if someone gets the private key, only one side of the message would get decrypted, since the keys used for encryption and decryption are not the same as in symmetric encryption process.

IV. OVERALL FUNCTIONALITY

- User can register to access the application.
- User can register with Single Sign On application using his email ID.
- User can register with File System using the same email ID
- User can login using the email ID in both the applications.
- If the user logs in to one application, in the server side of that application the last login time is checked.
- One server can generate the token for a user once the user logs in.
- Server generates the token if the token validity has expired.
- Server saves the IP address of the user machine and updates to the other server.
- Server encrypts the token using the email ID of the user.
- The first server updates the token to the other server, if the token is newly created for a user.
- The second server after receiving the new token checks the email ID
- Server updates the token if the user is registered.
- Server can decrypt the token and save the user IP address.
- User if access the URL from the same machine, user is directly redirected the user home page.

V. IMPLEMENTATION

Step-by-step processing of data

- The client logs into the applications.
- An HTTP request is sent to the server.
- The server then checks and retrieves the IP of the client machine.
- Generates token.
- Encrypts the token and sends it to the other server where the second application is running.
- The validity of the token is checked and the login time.
- The server sends an HTTP response.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

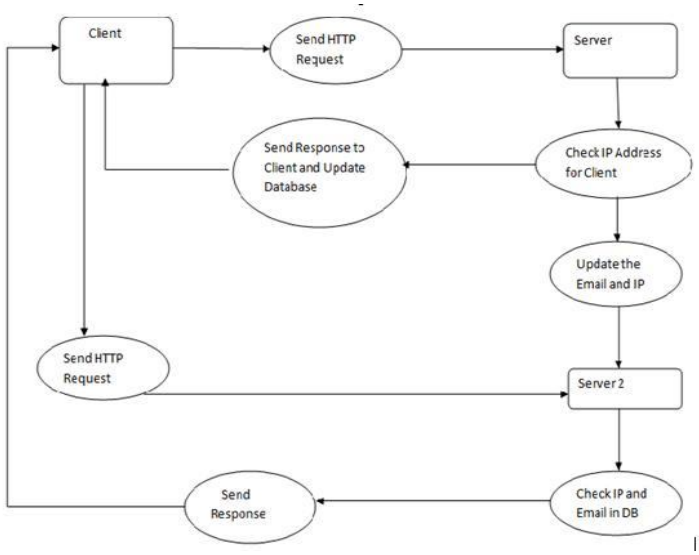

Figure 2: Data Flow

Discussion

A user can again log in into his account from a different system thereby its IP address in the database of the servers gets updated. The updated data is sent to the second server to keep both the servers in sync.

Once the token generated expires, on checking 24 hours from the time of login by the user, a new token gets generated for the user.

For the span of 24 hours the user will not have to enter his credentials again and can remain logged in onto his personal system. He has access to the services provided by both the applications namely the File System and the Mail System running on two different servers.

VI. CONCLUSION AND FUTURE WORK

Single sign on is basically as dealt with here the method to access multiple applications running on a distributed network with the help of a single log in credential. The user does not have to sign in repeatedly to access his account for the given time period of 24 hrs for which the user's credential and IP address of his system is valid.

Here we have dealt with the security issues involving Single sign on (SSO). So that when data is passed from one server to another it is in an encrypted format so that if intercepted security is not compromised. The second application on the second server updates its database in accordance with the data it receives from the first application and even checks the timer along with it which is set to 24 hours.

Thus an efficient and safe SSO method has been implemented.

In future we will try to integrate our Single Sign-On authentication scheme with different services. Like another server can be incorporated which has a cloud application like ownCloud running on it. Single Sign-On when implemented in connection with a cloud provides more flexibility and usability to the system. When connected to such an application it will provide a better and widespread use of storage and accessibility.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Here we have ensured security via symmetric key encryption namely Advanced Encryption Standard or AES. Some limitations had been encountered which can be overcome by using a different security mechanism. We can have a safer and better authentication scheme than AES which we have implemented here. Asymmetric key encryption can be used so that the key for encryption and decryption are different thereby increasing the security provided by the Single Sign-On mechanism. Another authentication scheme like attribute based encryption can also be used. It is also a public key encryption method but in this the key used for encryption is different than in AES. The key used in attribute based method as the name suggests is based on the attributes of the user. Thus few modifications in the future can make a more secure and flexible Single Sign-On mechanism to be implemented.

REFERENCES

[1] Arul Princy.A, Vairachilai.S, "A Survey on Single Sign-On Mechanism for Multiple Service Authentications", IJCSMC, Vol. 2, Issue. 12, December 2013, pg.40 – 44.
[2] Guilin Wang, Jiangshan Yu, and Qi Xie, "Security Analysis of A Single Sign-On Mechanism for Distributed Computer Networks", IEEE (Volume: 9, Issue: 1), Pg: 294-302.
[3] Chin-Chen Chang, Chia-Yin Lee, "A Secure Single Sign-On Mechanism for Distributed Computer Networks", IEEE (01/2012; 59(1):629-637. DOI: 10.1109/TIE.2011.2130500.
[4] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in Proc. of CRYPTO', 1993, pp. 232–249.
[5] W.B. Lee, C.C. Chang, "User identification and key distribution maintaining anonymity for distributed computer network", Computer Systems Science and Engineering 15 (4) (1999) pp 113-116,2000.
[6] T.S. Wu, C.L. Hsu, "Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks", Computers and Security 23 (2) (2004) 120-125.
[7] Cheng-Chi Lee, "Two Attacks on the Wu-Hsu User Identification Scheme", International Journal of Network Security, Vol.1, No.3, PP.147–148, Nov. 2005.
[8] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael"(PDF). National Institute of Standards and Technology. p. 1. Retrieved 21 February 2013.
[9] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.
[10] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures".
[11] Douglas Selent, "ADVANCED ENCRYPTION STANDARD", RIVIER ACADEMIC JOURNAL, VOLUME 6, NUMBER 2, FALL 2010. [12] Ritu Pahal , Vikas kumar , "Efficient Implementation of AES", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, pp 290-292, July 2013.

BIOGRAPHY

Noqaiya Ali is currently pursuing M.Tech. in Computer Science from PESIT, Bangalore, India. She received Bachelor's of Technology (B.Tech) Degree in Computer Science from ST.THOMAS' COLLEGE OF ENGINEERING AND TECHNOLOGY (STCET), Kolkata, India. Her research topic includes Security issues in Single Sign-On for Distributed Computer Networks.