Proceeding of the IEEE
International Conference on Automation and Logistics
Zhengzhou, China, August 2012

SAML-Based Single Sign-On for Legacy System

Fengming Nie, Feng Xu and Rongzhi Qi
College of Computer and Information
Hohai University
Nanjing, Jiangsu, China
nickel1217@gmail.com, njxufeng@163.com, rzqi@hhu.edu.cn

***Abstract* - The Single Sign-On is used for users to solve the logging in and the passwords managing problems in different application systems. However, the traditional SSO system cannot provide an appropriate solution for legacy systems which are independent and hardly modifiable. In order to solve the problem, we present a new SSO method based on the SAML legacy System. In this paper, the structure of the new method is given, the communication protocol between users and identity provider is defined, and the security of the method is analyzed. This method inserts the identity provider between systems and users without modifying them, authenticates the users by the SAML token, and implements the SSO in different application servers systems by auto form filling.**

Index Terms* - *Single Sign-On*；*Legacy System*；*SAML*；*Auto form filling

I. INTRODUCTION

As the increasing development of IT in different areas, the information systems are used more widely and can meet different needs from varies of areas, and the complexity of the systems is increasing. Normally, different systems have different development methods, different developers and different development phases, which would bring the following problems: users have to login and be authenticated many times if they want to use different systems; the more systems the task related to, the more complex relation the systems have, the more serious the problem. In order to solve these problems, the single sign-on (SSO) method is presented.

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. The advantages of single sign-on are obvious: the users do not need to remember all the usernames and passwords for each system, the time for switching the systems, logging in, and authenticating is reduced, the work for the administrators is simplified, and as the security is enhanced, the risk of cracking the password is reduced.

Based on these advantages, SSO is widely used in authenticating for developing the information systems now. However, there are still some problems when we put SSO into practice. For instance, some of the legacy systems in the application systems cannot be modified because the application systems are developed in an earlier time and the systems are independent to each other, so the users still need to enter their usernames and passwords to login these systems, which means SSO fails.

In this paper, we mainly focus on solving the problems in the SSO implementation in the legacy systems, especially in the web legacy systems. We try to modify the log-in method in SSO, so the identity authentication provided by the SSO servers is only used for determining if the user is permitted to log in the system when the user try to visit the Service Provider, while the user's role is still verified by the Service Provider. Based on this method, the SSO will reduce the workload of the system, and the difficulty of verification.

This method also lower the coupling between the user and the Service Provider, the user will only need to save the different log-in information for systems in the SSO server. The SSO will finish the log-in progress, turning the manual process into the automated process after the verification of SSO.

The rest of this paper is organized as: in Section 2, we give a brief introduction on the related works. In Section 3, we demonstrate the specific solution on SSO for legacy systems. In Section 4, we analyze the security level of the systems. And in Section 5, we give the conclusions.

II. RELATED WORK

The Kerberos-based single sign-on [1, 3], is authenticated by the electronic certificates provided by the centralized authentication servers. Though it is wildly used for its simple structure now, the legacy systems in it need great changes. The one time password method [2] is only suitable for the SSO without high performance requirements. The SAML (Security Assertion Markup Language) [10] is a standard based on XML, and is used for switching authentication and authenticating the data in different security domains. Steffen M. Hansen presents The SAML single sign-on protocol [6], and uses the static analysis to validate the SAML SSO protocol.

Takeshi [4] and Yihong Long [5] give two solutions on different legacy systems. Yihong Long's solution supports the automatic account bindings and the automatic application systems bindings. However, as the solution is restricted by the use of the web filter, and the SAML protocol is only used for the security transportation, the important data is still saved by the cookie and session, which makes the data still in dangerous. Takeshi's solution presents the LESSO model, which ensures the data security in transportation. However, it reduced the authentication efficiency.

III. METHODOLOGY

A. Model design

According to SAML framework, following components are needed to accomplish SSO:

(1) Identity provider (IDP)

978-1-4673-0364-4/12/$31.00 ©2012 IEEE

(2) The protocols for authentication information exchange among servers

(3) API among service provider (SP)

In this paper, we assume that the legacy systems cannot be modified. Instead of restructuring API, we login the application systems by filling forms.

We insert two services between the users and the Server Provider (SP), which are the SSO-Agent and the identity authentication server (IDA). These two services constitute the identity provider (IDP). The IDA is mainly used for issuing SAML authentication assertion token to users and identity users' identification when necessary. The user will provide the ID information (usernames and passwords) to the IDA, and submit it to IDA at the same time. After the user apply for SAML token from IDA, IDA will send the token to the user, and save the user's information in the service after the token is formed.

In a traditional SAML-based SSO system, the user requests the access to SP directly. However, in this paper, the user will login the SSO-AGENT, make a log-in request to SSO, when the user want to login the SP. The SSO-AGENT will verify the user's ID and the role, after the verification, the SSO will submit the form with the username and the password from original Service Provider. The Service Provider will respond to the user that if he is admitted or rejected to log in after conforming the information. We will show the details of this protocol in next section.

The model structure is given in Figure 1.

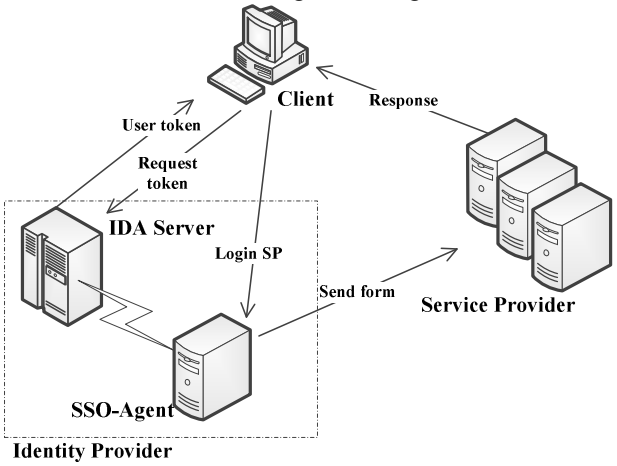

Fig. 1 Model structure

B. Protocol design

In this section we will illustrate the authentication communication protocol in detail.

(1) The user provides the ID information (username, password and etc.) to IDA, and asks for a SAML token.

(2) IDA verifies the user's information. If the information is valid, IDA generates a SAML token. At the same time, IDA encrypts user's information and saves it on the server.

(3) IDA sends the user-token to the user.

(4) The user logs in SSO-Agent, and applies to access a web service provider. In this step, the user agent issues an SSO-URL query, with the value of the encoded SAML token, to the SSO server at the identity provider. Meanwhile, the SSO-URL query has a RelayState parameter, which represents the URL or the unique code of the web service provider.
The RelayState parameter is a hidden parameter, and will send without modified or examined.

(5) After accepting the user's request, SSO-Agent asks for the user's cookie and gets the sessionid saved in the cookie immediately. Then SSO-Agent searches for the user's session object which matches to the sessionid on the server. If a valid session at the server already exists, SSO-Agent will send the information in the session to IDA directly to verify the user. If not, SSO-Agent needs to reextract the user credentials.

SSO-Agent decodes the user's request, and obtains the user token and URL of the service provider which user wants to access (the RelayState parameter), then sends these parameters to IDA to verify the user's identity.

(6) IDA gets the user token and performs a security check. While the token is verified to be safe and correct, and the user has a grant to access the service provider, IDA will select the user's username and password from the ID database at the server (The username and password are related to the service provider, encrypted through DES and saved in the ID database). After decrypts the username and password, IDA creates a SAMLResponse with the user's login information.
According to SAML2.0 standard, the SAMLResponse will have a digital signature which is encrypted based on IDA's RSA private key.

(7) IDA encodes the SAMLResponse, and returns it to SSO-Agent.

(8) While authenticating the SAMLResponse by IDA's public key, SSO-Agent acquires the user's login information from SAML assertion, and fills the form of the service provider automatically.

(9) SSO-Agent, instead of the user, request to log in the service provider with the user's information and the login form.

(10)The service provider verifies the login form, and discriminates the user's role of the service provider itself.

(11)Whether the login process is succeeds or fails, the service provider would response the final result to the user.

The specific workflow of the protocol is shown in the figure 2.

C. System implementation

The prototype system of this program is implemented in a LAN environment and a simulate network environment based on Java.

(1) ID provider

The SSO-Agent and IDA are built in the same service to reduce the communication risk, and also proved that this system could also occupy less resource of the system.

The interface of SSO-Agent, programed by JSP, provides the user with a unified login interface, makes the supported application service system list, and completes the operation of

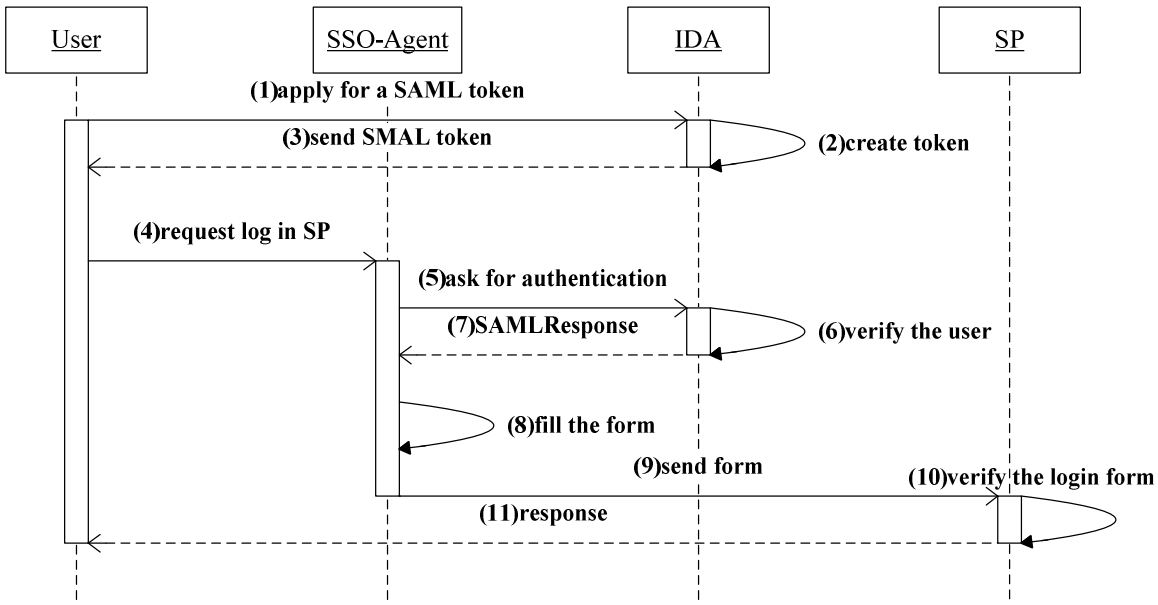

Fig. 2 Workflow of the protocol

the authentication of the user's username and passwords and the redirection of the website. The user could only log in other systems after the SSO login.

(2) Encryption and decryption module

This system needs a signature when the authentication server generates the assertions. The SSO agent needs to authenticate the information signature through the public keys of the authentication server. The communication between the user and SSO-agent needs to be encrypted to ensure the information safety. The JDK 1.6 from SUN comes with key generation package, and the keytool in JDK can be used to generate the keystore. The encryption and decryption can be done by the encryption code, the decryption code and the signature code in the backstage.

(3) The Communication

The SAML data should be transported by SOAP, which means the encrypted information would be SOAP packed and unpacked, and the packed and unpacked programs are also programed by JAVA.

IV. Security analysis

As each application service in this system is independent to each other, security of the service provider and the access control of users are mainly depended on their own servers. For that reason, 3 aspects of should be put into consideration when we analyze the security of this system. They are the ID provider, the client and the communication.

A. ID provider

In the ID provider, the attacks mainly focus on the SSO-agent and the authentication services. In the SSO-Agent server, the user's session-object should be saved so the user could be re-authenticated. The user's information in the session-object is encrypted by the public keys, so the security of it mainly depends on the security of the encryption algorithm. In the authentication server, the user's SAML token data and the user's username and password mapping in other application systems are saved in the database, so the security of it mainly depends on the security of the database. The security of the database is acceptable for the data in the database are encrypted.

Another effective attack for the ID provider is the replay attack by intercepting the user's session-object. So in some circumstance, the session-object signature verification is necessary. And when the security demand is very high, the session-object should not be saved, while we can have the high security by reducing the efficiency.

B. Client

The weakness of client security is the user's username and password. Using a keylogger, the attacker can get the typed username and password when the user tries to log in the SSO-Agent. For this reason, the security of the client depends on the user's sense of security, can get typed username and password when user is logging in the SSO-Agent. For this reason, the security of the client depends on user's sense of security.

The hardware methods, rather than the software methods, can help us to achieve a higher security level, such as fingerprint scanning or smart cards. As for the attack against the client, the hardware methods can reduce the possibility of password-cracked effectively. As a result, the attacker can't log in the SSO-Agent without knowledge of user's hardware information. On the other hand, the authentication hardware must be used on the client, which may bring the client the more workload.

C. Communication

This system mainly includes three communication modules:

(1) The communication between users and IDP;

(2) The communication between SSO-Agent and IDA in IDP;

(3) The communication between IDP and SP.

In the situation (1), the user needs to submit the ID information to identity provider, and provide the user token to SSO-Agent when login. Because the communication is processed in SSL, it is safe if the SSL communication is secure.

In the situation (2), the SSO-Agent and IDA are built on the same server in prototype system, can contribute to decrease the risk of communication. If SSO-Agent and IDA are built on the different servers, they change the user authentication through SAML assertion, so it's necessary to ensure the security of the SAML message. We must append a digital signature to those messages. In addition, adding the timestamp into SAML messages can evades the DoS attacks and replay attacks. The communication also can be processed in SSL to ensure a further safety communication.

In the situation (3), as the legacy systems cannot be modified, they cannot recognize the form if we only encrypt the forms in the SSO. So it is the most vulnerable part in the system. The security level when the ID provider submits the user's login form should be discussed according to the differences between service providers. For those service providers which are not necessary to encrypt the forms, we can only frequently change the password and update the legacy system as fast as we can to make sure its safety.

V. CONCLUSION

In this paper, a solution on SAML-based SSO for legacy system is presented and discussed. Firstly, we do some research on the related works about SSO, introduce the useful technologies relevant to the paper and analyze the deficiency of resolution hitherto devised. Secondly, we present the system model and communication protocols, design the SSO-agent and IDA, and insert them between the users and application servers seamlessly. Thirdly, after the security analysis, we can prove that this solution can ensure the efficiency and protocol security.

In the future, we will focus on the study of the system security. We will try to improve the security level by studying the hardware security technologies, such as the trusted computing platform.

REFERENCES

[1] Neuman.B.C. and Ts'o. T. "Kerberos: An Authentication Service for Computer Networks" ,IEEE Communications Magazine, vol. 32(9), pp. 33-38, Sept. 1994.

[2] Tiwari.P.B and Joshi.S.R. "Single Sign-on with One Time Password", 1st South Central Asian Himalayas Regional IEEE/IFIP International Conference on Internet, AH-ICI 2009, Nov. 2009.

[3] SHEN Yang and DU Zhong jun, "Research and design of single sign-on based on Kerberos protocol", Computer Engineering and Design, vol. 32(7), pp. 2249-2251, July 2011.

[4] Nishimura. Takeshi and Sato, Hiroyuki, "LESSO：Legacy Enabling SSO", Proceedings - 2008 Inter. Symp. on Applications and the Internet, pp. 301-304, 2008.

[5] LONG Y.H,LI C.Y.,TANG Z.H. and LIU Xu, "A Transparent Single Sign-on Solution for Web Legacy System", Information Security and Communications Privacy, pp. 67-69, Oct. 2010.

[6] Hansen S.M. ,Skriver Jakob and Nielson H.R., "Using static analysis to validate the SAML single sign-on protocol", Proceedings of the 2005 Workshop on Issues in the Theory of Security, WITS '05, pp. 27-40, 2005

[7] Peng, Jiezhao and Wu, Qi, "Research and implementation of RSA algorithm in Java", Proceedings - International Conference on Management of e-Commerce and e-Government, ICMeCG 2008, pp. 359-363, 2008.

[8] Yuen-Yan Chan, "Weakest Link Attack on Single Sign-On and Its Case in SAML V2.0 Web SSO", Lecture Notes in Computer Science, vol. 3982/2006, pp. 507-516, 2006.

[9] IETF, http://www.ietf.org/rfc/rfc1321.txt

[10] Organization for the Advancement of Structured Information Standards(OASIS), "Assertions and Protocols for the OASIS Security Assertion Markup Language(SAML)V2.0"[EB/OL], http://does.oasisopen.Org/security/saml/v2.0