A Conceptual Framework Secure Web Service: Secure Transaction Logging System

Nor Izyani Daud and Khairul Azmi Abu Bakar
Information System Security Lab, MIMOS Berhad, Kuala Lumpur, Malaysia
Email: {izyani.daud, mie}@mimos.my

***Abstract*—Web service is one of the technologies that we can use to build a web application nowadays. However, there are many potential treats that can cause a vulnerability to web service if we didn't develop in securely. This paper provides an approach on how to develop a secure web service application with mutual authentication technique. In this research, we are using certificate as our authentication method. We also add an extra security measure where it provides a database secure for the application. This paper starts with an introduction about the security and an overview about web service technology. If follows with current issue of web service. Some other different approaches on how to develop a secure web service are discussed in this paper. Author will elaborates in detail about the propose system; secure transaction logging system. It is includes system architecture, process flow, implementation and deployment of secure transaction logging system.**

***Index Terms*—log system, secure system, secure web service, web service technology**

I. Introduction

How secure is your system? What are the methods you are using to ensure that your system is secure enough for any online transaction? Do you need to have a powerful authentication system to ensure your system is well protected? These are the question that you will be asked when you are building an application.

Mano Paul quoted in a book "The 7 Qualities of Highly Secure Software" [1], over 535 million sensitive data record were exposed by more than 2000 attacks. There are over 310 million out of 535 million data records were breach as a result of hacking. This figure is based on the chronology of data breaches, published by Privacy Rights Clearinghouse since 2005.

As reported by General Incident Classification Statistic data from MyCert Malaysia in 2014 [2], the current total number of incident reported for year 2015 as the current date is 7399 cases. An article titled "Malaysia is sixth most vulnerable to cybercrime" [3], stated that Malaysia was in the list along with several countries in the Sophos Security Treat Report 2013.

From these two cases, it shows that security of system is an important aspect that we need to consider when developing an application. By developing secure software, you can also minimize the security treat.

There are several ways we can do to ensure that we develop secure software. As quoted in a book titled "Securing Information and Communication Systems: Principles, Technologies and Applications" [4], a comprehensive security solution should involve recommendation and countermeasures in area; which are technical, physical, procedural, legal and personnel.

One of the technologies that we can use to develop a system is by using a web service. According to Wikipedia [5], a web service is software designed to support interoperable machine-to-machine interaction over a network. Client will invoke the web service when they are trying to get access to the application. The examples of technology that we can use to design web service are by using XML (eXtensible Markup Language), WSDL (Web Service Description Language) and SOAP (Simple Object Access Protocol). However, most of the web services still have security issue.

In the next chapter of this paper, author will discuss about the current issues in web service. Next, techniques or methods that are implemented by others to secure the web service are discussed. Author than will elaborate in detail, the technique that we proposed for securing the web service. It is also includes software requirement and configuration for the system. Security features that author add in the system also will be explained in this paper.

II. Current Issues

Web service is design to ease the delivery of service over the Internet. Because of this, it has become a popular technology that people use for their application. However, people still forget that there some security issues about the web services. Basically, we conclude issues with the web services security in two main areas.

A. No Authentication and Authorization to Access the Web Service

If there is no authentication and authorization whenever user wants to get access to the web service, it means that the web service is able to be accessed by anyone within the boundary. This is not a good practice where there is no control or protection to the information in the system.

From journal titled "Issues and Challenges in Web Service" [6], it described that one of the issues about web

Manuscript received February 10, 2016; revised June 17, 2016.

doi: 10.18178/lnit.4.1.25-29

service implementation is un-authorized access. In most of the web service implementation, there is no authentication or authorization needed when user wants to get access or invoke the web service. It allows outside users to get access to the database and application.

B. Un-Encrypyted and Un-Secured

Un-encrypted and un-secured web transaction also may cause an issue to the web service application. David Geer quoted in this article titled "Taking Steps to Secure Web Service" [7], the un-encrypt and un-secured web service transaction creates the potential disaster.

Every transaction through Internet is recommended to use encrypted channel to ensure that the transaction is not tampered by anyone. In encryption channel, the information may be accessed by un-authorized user by changing value in the HTTP header. The connection between browser and the server can be secured by using techniques such as SSL encryption.

III. Related Work

There are several techniques that have been proposed and discussed by others to secure web services. From paper titled "Secure E-business Transactions by Securing Web Services" [8], it quoted that .Net technology provides very good security features to the Integrated Development Environment (IDE). IIS plays a very crucial role in web applications and web services. The security mechanism thot provided by Internet Information Services or IIS can be grouped into for basic categories; logging, fault isolations, access control and message protection. This paper also suggests that web service can be secured by providing secure communication using SOAP. Other than that, there are various options from protocol based, platform based or message based security that developer can choose to ensure their web services is secured. To perform this, system can limits the number of users who can get access to the service and grant permission from only certain IP address to get access to the service.

Ying Lui, Tet H. Yeap and William O'Brien [9] proposed adding XML Web Services security component to an existing Virtual Private Network (VPN) gateway to provide integrated security solution for both XML Web Services and traditional network-based applications. To add more security to the current approach, both XML Web Service security and VPN server on the gateway are sharing the same digital Elliptic Curve Cryptography key.

From paper titled "Architecting secure Web Services through policies" [10], suggested to use policy as a security solution for the web service. In the implementation, each application and web service will be bound to a XML based policy file. This policy devoted to the task; where it can sign and encrypt or decrypt and check the inbound called whenever the web services are invoked.

As quoted by Rattikon Hewett and Phongphun Kijsanayothin in their paper titled "On securing privacy in composite web service transaction" [11], the authors highlighted privacy issue in web service transaction. The authors provide an intelligent semi-automated privacy-aware approach to efficiency building an appropriate composite web service that satisfies functional requirements and complies customer privacy preferences and trust.

IV. Proposed Approach

A. About the System

Secure transaction logging system is a system that provides secure transaction when user wants to log a transaction between two machines. The system is using web services technology. To provide security to the system, we use mutual authentication by using certificate for SSL connection.

Mutual authentication is a security feature which a client must prove its identity to a server and the server must prove its identity to the client before the transaction happen.

Secure logging system is designed to provide a secure database. Database mode is change to read-only. The system restricted any user or system administrator to make changes to the database. The option to delete or update record and table option is disabled. The system only allow registered user to log in to the database. Furthermore, system only allows owner of the data to view their own information. To protect the user credential, the system also performs hash function to the user identity value. Since in this system, the author used certificate as a user token, the user identity value is Distinguished Name (DN). Distinguished Name is a string represent the uniquely identity of the user. Fig. 1 shows DN for a user in the certificate.

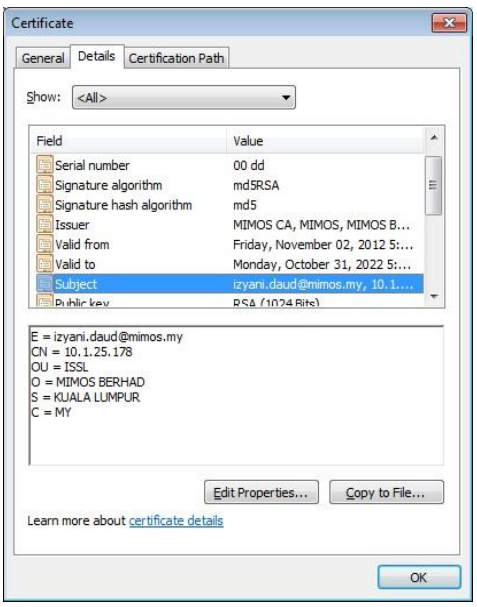

Figure 1. User distinguish name example.

In the above example, the values for DN in this example are email, common name, organization unit, organization, state and country. In this implementation, the system use IP address of the machine as their common name.

B. System Architecture

Secure transaction logging system consist of two machines that communicate using web services through SSL secure channel. In current implementation, system is using a certificate as user credential. Server also requires having their certificate as their own credential. During handshaking process, both user and server need to present their own credential to authenticate themselves to get access to the system. In this system, we considered a user as machine that is trying to get access to the information from the server. Fig. 2 illustrated system architecture for secure transaction logging system.

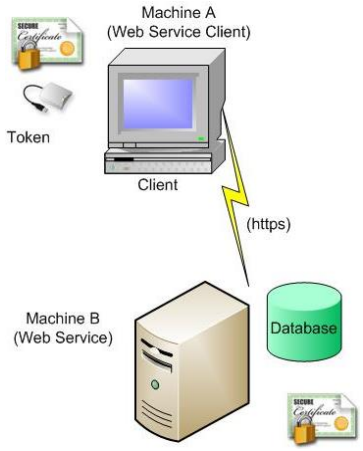

Figure 2. System architecture.

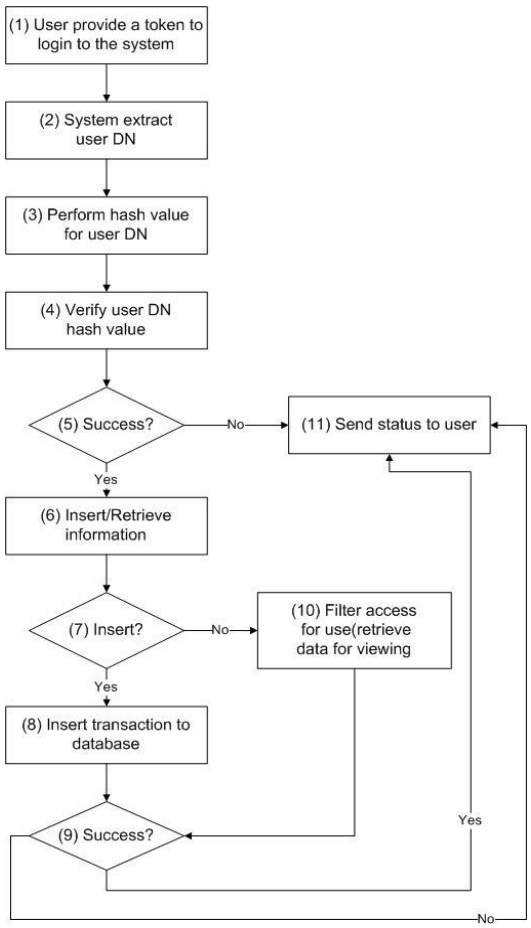

Figure 3. Secure transaction logging system flowchart.

C. System Flowchart

Fig. 3 describes the flowchart of secure transaction logging system. In current implementation, the token used is user certificate. The system was designed to log transaction and view information from the database.

The system starts when user is trying to access the web service. Since it mutual authentication, both user and server need to present their own credential. Once handshaking process successful, system extracts user Distinguished Name (DN) from the certificate. System then hashes the user DN value and compares it with the information that already stored in the database. If it is matched, system will allow user to store information to the database. Users are only allowed to access information that belongs to them.

However, there is a pre-registered process before user can use the system. User need to store the user DN value to the database before use the system.

D. System Implementation

This chapter discuss about the software requirement and configuration that needs by the system. The system is build using Java language.

1) System requirement

We use MySQL and Apache Tomcat to build the system. The system also requires having certificate file for client and user. In our current development, we are using server trust store and server key store to store the certificate value.

2) Configuration

There are some configurations that we change to enable the secure transaction logging system. For MySQL, we need to enable read-only mode by changing database type to ARCHIVE mode. For Apache Tomcat, we need to configure server.xml to enable mutual authentication for the system. The example of the configuration is shown in Fig. 4.

```
<Connector port="8443" protocol="HTTP/1.1"
SSLEnabled="true" maxThreads="150"
scheme="https" secure="true"
clientAuth="true" sslProtocol="TLS"
keystoreType="JKS"
keystoreFile="C:\serverWS.keystore"
keystorePass="123456"
truststoreFile="C:\serverWS.truststore"
truststorePass="123456"/>
```

Figure 4. Apache configuration.

3) Deployment

This chapter discuss about the software requirement and configuration that needs by the system. The system is build using Java language. In this implementation, the author creates two separate programs for secure transaction logging; one is to be installed in client side, one is to be installed in server side. For program that needs to be installed in server side, author only need to import the web service to Apache Tomcat Server, server key store and trust store also need to be installed in the server. For program that needs to be installed in client side, author build the program library format file. Every user that wants to use the system needs to call the library.

There also needs to specify their client certificate by indicate the client key store and trust store. Fig. 5 is the information that you need to specify when you are calling web service application from your Netbean IDE.

```
-Djavax.net.ssl.trustStore=c:\\client178.truststore
-Djavax.net.ssl.keyStore=c:\\client178.keystore
-Djavax.net.ssl.keyStorePassword=123456
```

Figure 5. IDE configuration.

E. Security in the System

When we designed the system, we wanted to include as much security features. In this system, we successfully embedded four security features. The features are:

1) Encrypted and secure transaction

The system uses mutual authentication by means of using certificate to authenticate both user and the server. The client and server need to present their own credential during handshaking process. Using SSL and mutual authentication, system can ensure that only the authenticated and authorized users have the permission to access the system.

2) Provide authorization

Secure transaction logging system provides a strong authorization. The system only allows owner of the data to view information that belongs to them. Furthermore, when user wants to log their transaction to the database, the system will check and ensure that the current user has already registered to use the application. The identity of the user is extracted from the presented certificate.

3) Database security

The system is also designed to provide a database security. In current system, we already changed the database mode to read-only. No one is allowed to make changes or delete the information that had been stored to the database.

The system also checks that only authorized user are able to log their information to the database. To provide the privacy of the information, only owner of the data are able to view the information that belongs to them.

4) Hash function

Hash function is a method where we transform the original message or value to something else. The purpose of the method is to provide protection to the data; where it makes people from outside hard to get the real value of the data. An example of hash is function SHA1.

In our system, we use hash function to user Distinguish Name. It means that when user login to the system, the system extracts and hashes user DN before insert the information to the database.

V. Conclusion

In this paper, we propose secure web service technology with mutual authentication using certificate for our secured logging system. We also provide an authorization process before the user is allowed to view and insert the data to the database. To protect user data, we use hash function for user Distinguish Name. Database used also had been changed to read-only mode.

From related works, we found that most web services do not use secure connection. There are proposing some other technique for example VPN technology.

Technology that mentioned here is not the best technology. In future, the technology that we mentioned here can be improved by adding some of other build-in technology for example using SAML. This might help the security of web services become much better compare to the current implementation.

Acknowledgment

We would like to thank to the team leader for the support in order for us to develop secure transaction logging system. Without guide from them, it is very difficult for us to complete the system. By having the system, we are able to explore more about securing web service technology.

References

[1] P. Mano, *The 7 Qualities of Highly Secure Software*, 1st ed., Auerbach Publications, 2012.
[2] MyCERT incident statistics. [Online]. Available: https://www.mycert.org.my/statistics/2015.php
[3] Malaysia is sixth most vulnerable to cyber crime. [Online]. Available: http://www.thestar.com.my/News/Nation/2014/09/23/cyber-crime-malaysians-sixth-most-vulnerable/
[4] J. Lopez, S. M. Furnell, S. Katsikas, and A. Patel, *Securing Information and Communications Systems: Principles, Technologies, and Applications*, 1st ed., Artech House, 2008.
[5] Web service. [Online]. Available: https://en.wikipedia.org/wiki/Web_service
[6] S. Tanwar, "Issues and challenges in web services," *International Journal of Advanced Research in IT and Engineering*, vol. 1, no. 2, 2012.
[7] D. Geer, "Taking steps to secure web services," *Computer*, vol. 36, no. 10, pp. 14-16, Oct. 2003.
[8] A. T. Siddiqui and A. K. Singh, "Secure E-business transactions by securing web services," in *Proc. International Conference on Management of e-Commerce and e-Government*, Oct. 2012, pp. 79-84.
[9] Y. Liu, T. H. Yeap, and W. O'brien, "Securing XML web services with elliptic curve cryptography," in *Proc. Canadian Conference on Electrical and Computer Engineering*, April 2007, pp. 974-977.
[10] M. Mashood and G. Wikramanayake, "Architecting secure web services through policies," in *Proc. International Conference on Industrial and Information Systems*, Aug. 2007, pp. 5-10.
[11] R. Hewett and P. Kijsanayothin, "On securing privacy in composite web service transactions," in *Proc. International Conference for Internet Technology and Secured Transactions*, Nov. 2009, pp. 1-6.

Nor Izyani Daud was born in Kuala Lumpur, Malaysia. She received the B.A Hons Degree in Information Technology, Artificial Intelligence majoring from the Universiti Utara Malaysia in 2000; and Master Degree in Real Time Software Engineering from Universiti Teknologi Malaysia in 2006. In 2006, she joined MIMOS Berhad as a Senior Engineer. She has been working in Information Security areas; for example smart card programming, web development technology, networking are and security scanning and analyzing. She also involved with CMMI-Capability Maturity Model Integration implementation in the organization.

Khairul Azmi Abu Bakar received the degree in Computer Engineering from Iowa State University, USA in 1995 and master degree in Communication and Computer from National University of Malaysia in 2002. He was awarded Ph.D. degree in Electrical Engineering from University of Strathclyde, United Kingdom in 2012 for the study on free-riding nodes in an open MANET. He is currently a staff researcher at MIMOS Berhad where he has been since 1996. He has been involved in several R&D projects in the field of micro-controller, smartcard, security systems under open source platform. His primary research interests include wireless ad hoc security, authentication system and computer network.