2012 International Conference on Information and Computer Applications (ICICA 2012)
IPCSIT vol. 24 (2012) © (2012) IACSIT Press, Singapore

Controlling Web Services Access through a Trust based System

[+] Rohit Dhand and G. Geetha

Lovely Professional University PUNJAB – India

Abstract. Service Oriented Architecture has emerged as one of the main standards of writing software in the modern era. Whether we are writing web based apps or web services, SOA has started dominating the trends of software development. With the rapid growth of Web 2.0 and Cloud Computing, SOA has emerged as one of the industry accepted norms. With the growing demand of Cloud Computing and ever increasing implementation of Web Services, problems of authorization control of these web services is also shooting up at a phenomenal level. With numerous web services hosted up, it would become a cumbersome process to manually provide access control to the users by the system administrators. This paper presents a concept of Trust based system than can help in access control of different web services.

Keywords: Web service, WSDL, callback, request-response, Trust, Framework.

1. Introduction

Web Services have given Enterprise computing a new dimension for performing business activities. [1,2] They provide interoperability between numerous business processes and also bring automation of these tasks for various organizations. All the Web Services are based on the fundamental principle of Service Oriented Architecture. The model of SOA is changing the way of writing business and web applications. With SOA more and more organizations are redefining the existing architecture for new applications and support for legacy applications as consumable services. Service Oriented Architecture has the potential to make the concept of reusability real. There have been lot of talks over the years about reusability and its implementation, but nothing concrete has been done to convert this concept into a reality. Typically an IT manager in any business house does not know how many duplicate processes are being run in an ERP system. So by using Service Oriented Architecture the business houses can identify the overlapping processes in the system. The following gives an example of overlapping processes.

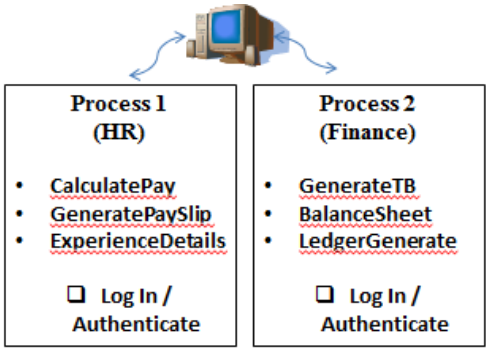

Figure 1.Duplicate Log In / Authenticate Process

[+] Corresponding author.
E-mail address: dhandrohit@gmail.com; gitaskumar@yahoo.com

Figure 1 shows how the login facility is duplicated in the system. The organization is paying multiple times to get same functionality in different processes. But this can be changed with SOA implementation as shown in Figure 2

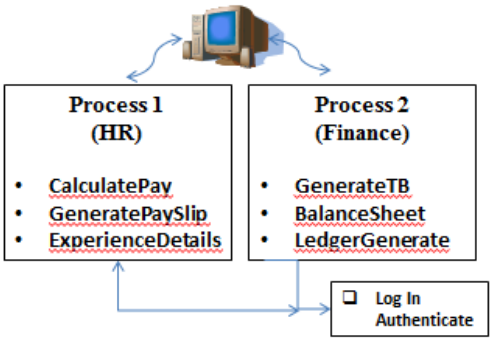

Figure 2 LogIn/Authenticate Process through SOA implementation

Figure 2 clearly shows the login service now implemented in the form of a reusable component. Different services of HR, Finance are using common function of Log In/Authenticate. So the organization is saving in terms of cost and implementation to get the functionality of authentication.

2. Web Services

Service Oriented Architecture is basically development and consumption of Web services [3]. Web Services are simply a piece of logic/code that runs on Web Server and handles the requests of multiple clients. These Web Services are also trusted because they communicate using standards-based Web technologies including HTTP and XML-based messaging [4,5]. They are designed to be accessed by other applications. They can perform operations such as checking accounts balance, CRM, SCM etc. Web services are purely independent in terms of hardware and software. This means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranet or internet using Web services. Web services are powered by the following core components:

i. XML
ii. WSDL
iii. SOAP
iv. UDDI **[6]**

With large deployment of Web Services in today's era on multiple platforms, there arises a problem of creating a reasonable combination of accessibility and access restrictions [7] among the administrators. The access to the web services must be controlled and authorized because of the following:

i. More and more web services are written and they can be misused by an unauthorized client/user.
ii. Web Services normally access the databases at the back end. This may involve content and confidentiality based approvals.
iii. Employing an IT administrator to do the task of administration of Access Control looks, out dated and cannot replicate in the large setup where number of users and resources to be configured are in a big volume.

So the access control on web services managed by the System administrator(s) will look out to be a tedious, complex and a mammoth job in the near future. This generates the demand of the system that can automate the process of authorization control to an extent so that the system itself takes care of the majority of the authorization jobs. But this does not mean that the human intervention will go away completely. It will always be needed to provide another layer of flexibility. There can be a semi-intelligent system that works on the mechanism of **"**Trust**"** to automate the task of access control.

3. Meaning of Trust And Formal Representation of Trust

Trust [8], is a very important aspect of human life. For e.g. we trust the police for our security. Trust is core to all transactions. When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high. Correspondingly when we say that someone is untrustworthy, we imply that the probability that he will perform an action that is beneficial to us is very low.

In fact the trust can be represented within a range from 0 to 1, where 1 represents pure trust and 0 represents distrust. Trust can have the following characteristics:

i. Trust is relativized to some transaction. A may trust B to drive a car but not to sell the car.
ii. Trust is a measurable concept. A may trust B more than A trusts C for the same action.

Trust relationships are usually based on objects, services, and confidentiality of the content factors. In order to make the concept clearer, we can formally represent the above in the following manner:

i. X and Y can be treated as objects of a class named **Person.** They can also be called as Requestors **[9].** Every object of class Person has an initial Trust value represented as **Tv(X). If Tv(X)=1** then it means that the system trust on X is blind or complete.
ii. X can trust Y which is represented by **Tv(X)->Y. If Tv(X)->Y=1** then it means that X has a complete trust on Y.
iii. In any system the trust will be based upon some set of services belonging to any class. For e.g. Human resource can have different services like: Generating Experience Certificate, Issuing duplicate Salary Slip etc. **So Tv(X, service)->Y=1 means** that X completely trust Y for a particular service.
iv. Every service in a system accesses some data/content which has got some confidentiality factor. The content with its confidentiality factor can be established as:
CCi(service). If CCi(service)=0 it means the content with its confidentiality factor is of least importance. Anybody can access that. If CCi(Service)=1 then it means that the content with its confidentiality **factor is of maximum importance. Only restricted users or users with high roles can access the same.**
v. Any service can take recommendations from other set of services within the system or outside the system. This is called as **recommended trust** and is denoted by **RTv(X).** For e.g. if X is accessing any service of HR, then the Central Trust Authority can take recommendations for X from other departments like Marketing and Finance and it can be used towards generating its **Trust Value** which can be represented as:

$$RTv(X) = \frac{Tv(X, ServiceMar) + Tv(X, ServiceFin)}{2}$$

vi. For every action of the object like X, if the service is granted access, then the object X will get an appreciation in terms of updating his **TrustValue** represented as:

$$Tv(X) = Tv(X, Service) + CCi(Service)$$

Similarly if X doesn't get the access to the service, then the he gets punishment in form of decrement in his **TrustValue.**

$$Tv(X) = Tv(X, Service) - CCi(Service)$$

In order to understand the above formal representation, let's assume X is going to access the service of generating duplicate pay slip from HR Services. So his trust value will be calculated as:

$$Tv(X) = Tv(X, S1) + \left[Tv(X, S1) * \left(Tv(X, S1) \right) - CCi(S1) \right]$$

Similarly if the recommended trust is available from other departments like Finance and Accounts, then we will calculate the **RTv(X)**

$$RTv(X) = \frac{Tv(X, ServiceAcc) + Tv(X, ServiceFin)}{2}$$

Based upon the two equations, the Trust Value of X is calculated as:

$$Tv(X) = RTv(X) + \left\{ Tv(X, S1) + \left[Tv(X, S1) * \left(Tv(X, S1) \right) - CCi(S1) \right] \right\}$$

Now if the calculated **Tv(X)** is equivalent or greater than the TrustValueIndex required for the service to be accessed, then X will earn an increment in his trustvalue index which is denoted by:

$$Tv(X) = Tv(X, Service) + CCi(Service)$$

Otherwise X is entitled to punishment which will be denoted by:

$$Tv(X) = Tv(X, Service) - CCi(Service)$$

4. Conclusion

Automated authorization control is an untouched concept. Organizations were not investing heavily in this area. But now as the deployment of the web services is growing tremendously, we need to think about a system then can reduce the human intervention of access control. Trust based system is one of the many approaches to achieve the same. This paper introduces the key concepts based on 'Trust' for ensuring control on the authorization of the web services on role based model. This model can be implemented in any kind of framework and also on any kind of organization.

5. References

[1] Jinpeng Wei, Lenin Singaravelu and Calton, "A Secure Information Flow Architecture for Web Services Platforms", IEEE Transactions on Services Computing, Vol. 1, No. 2, April – June 2008.

[2] Laurel Reitman, James Ward, Jack Wilber, "Service Oriented Architecture (SOA) And Specialized Messaging Patterns", A technical White Paper published by Adobe Corporation USA, 2007

[3] Erin Cavanaugh, "Web services: Benefits, challenges, and a unique, visual development solution", A white paper published by Altova, Inc. USA, 2006.

[4] Z. Song, S. Lee and R. Masuoka, "Trusted Web Service", Pro. Second Workshop Advances in Trusted Computing, 2006.

[5] M. Turner, et al. "Using Web Service Technologies to create an Information Broker". In *Proceedings of the IEEE International Conference on Software Engineering (ICSE)*, pp. 552-563, 2004.

[6] J-Y Chung, K-J Lin, R.G. Mathieu. "Web Services Computing", In *IEEE Computer* Vol. 36, Issue 10, pp 38-44, Oct. 2003.

[7] Shiping Chen, John Zic, Kezhe Tang, David Levy, "Performance Evaluation and Modeling of Web Services Security", IEEE International Conference on Web Services ICWS, 2007.

[8] Dhafer Thabet, Lamia Hassine, Henda Ben Ghezala, "Situational Secure Web Services Design Methods", International Conference on Software Engineering Advances, 2007

[9] Dhafer Thabet, Lamia Hassine, Henda Ben Ghezella, "Toward Situational Secure Web Services Design Methods", IEEE Transaction on Service Computing, 2008.