Web Services and Security

Vaibhav Ingle, Nilesh Swami, Mahesh Shelke, Saurabh Kataria, Chhaya Varade

Abstract— The vision of a landscape of heterogeneous web services deployed as encapsulated business software assets in the Internet is currently becoming a reality as part of the Semantic Web. When pro-active agents handle the context-aware discovery, acquisition, composition, management of applications services and data, ensuring the security if customers data become a principle task. In this paper we propose neoteric way web services and security. A methodology based on type-based information flow to control the security of dynamically computed data and their proliferation to other web services. The approach is based on the following trine guidelines: (1)The business and security concern of integrated web services are separated and building them independently.(2)Runtime modification of integrated web services.(3)Providing compartmentalization so that one service can not affect another. We are developing flight system to demonstrate the feasibility of our approach.

Index Terms—The business and security concern of integrated web services are separated and building them independently, Runtime modification of integrated web services, Providing compartmentalization so that one service can not affect another.

I. INTRODUCTION

As we all know that the security is must for any valuable thing and hence the topic of security is trending nowadays. In this situation clients consider security to be delivered immediately even on programs that were not developed with security in consideration. When the systems are to be developed for the web/networked environments the challenge is even competent.

A web service [1] is a standards-based, language-agnostic software entity that accepts specially formatted request from other software entities on remote machines via vendor transport neutral communication protocols, producing application specific responses.

The simplest web service system has two participants :

(i) A service producer (Provider).

(ii) A service consumer (Requester).

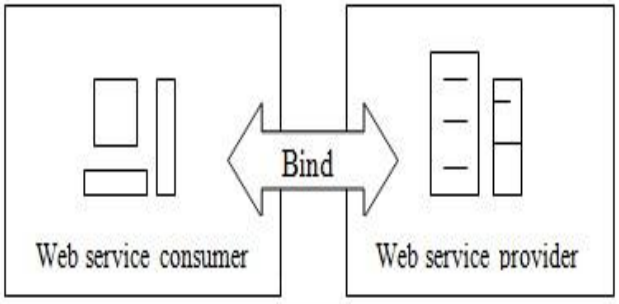

Fig 1 : Simplest web service system.

The successful deployment of this technology cannot hide the security breaches and threats that a web service can be exposed to. Enforcement of web services security is one of the most important duties, which the research community has to perform. This means the design and development of concepts, processes and tools that help in making Web services protected from malicious users. Instances of these security features can be the enforcement of user authentication, access control and data confidentiality in web services. Varietal standard languages have been determined to apply Web services security. The Security Assertion Markup Language (SAML) [2], WS-Security [3] and WS-XACML [4] are the utmost effectual ones. The prime concern with language based strategies is that the prime aim is on embedding the security features in the design/code of the Web services, i.e., integrating them statically. However, many security features require run-time verification of the security policies, which may often be modified and updated. This means that when the security policies, and/or the verification strategy change, the developer has to go back to the design/code of the Web services and update them accordingly. This problem is raised more when several Web services are composed together in a BPEL [5] (Business Process Execution Language) process to form a more complex system. With the use of the BPEL, there is a lack of modularity for modeling cross-cutting concerns in the environmental and addition or removal of partner Web services requires static and dynamic adaption.

In other words, if a BPEL runtime process change is required, we have to stop the running process, change the needed Web services, modify the composition and then restart. Such a mechanism is cumbersome, error-prone and tedious.

We propose, in this paper, an aspect-oriented approach for the necessitation of Web services security. Our hypothesis is based on a union between Aspect-Oriented programming (AOP) and integration of Web services. AOP allows to specify the security concerns in separate components called aspects. These aspects are then integrated in the BPEL process at runtime.

To validate the feasibility of our proposition, we developed a Flight System (FS) that is composed of several Web services. A RBACK (Role Based Access Control) model for the flight system, which we called RBAC-FS, is elaborated.

Manuscript Received on May, 2014.

Vaibhav Ingle, Information Technology, University of Pune/ G.H.Raisoni Institute of Engineering and Technology, Pune, India.

Nilesh Swami, Information Technology, University of Pune/ G.H.Raisoni Institute of Engineering and Technology, Pune, India.

Mahesh Shelke, Information Technology, University of Pune/ G.H.Raisoni Institute of Engineering and Technology, Pune, India.

Saurabh Kataria, Information Technology, University of Pune/ G.H.Raisoni Institute of Engineering and Technology, Pune, India.

Lect. Ms. Chhaya Varade, Information Technology, University of Pune/ G.H.Raisoni Institute of Engineering and Technology, Pune, India.

Afterwards, the Web services that implement the security features are developed. Then, the BPEL aspects that integrated the security functionalities dynamically into the BPEL process created. The devised aspect realize the elaborated RBAC-FS model and provide authentication and access control features to the flight system. Case studies and experimental result are also presented to defend our propositions.

II. RELATED WORK

The research community is fascinated towards the Web services security topic. From the definition of standards to the publication of research papers, the goal is to provide policies and mechanism for enforcing web services security. In this context, several standards such as Security Assertion Markup Language (SAML), WS-Security and WS-XACML were considered.

The Security Assertion Markup Language (SAML), developed by the Security Services Technical Committee of OASIS, is an XML-based framework for communicating user authentication, entitlement, and attribute information. As its name suggests, SAML allows business entities to make assertion regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application. SAML can be used to manage secure sessions between organizations and can leverage several mechanisms such as basic password authentication, SSL and X.509 certificates, etc.

A security token is delivered to the requester after successful authentication. This security token allows granting certain permission to the requester. It should be noted that the story of SAML need not end with its published set of assertions, protocols, bindings, and profiles. It is designed to be highly flexible, and thus it comes with extensibility points in its XML schemas, as well as guidelines for custom-designing new bindings and profiles in such a way as to ensure maximum interoperability.

WS-Security protocol was originally developed by IBM, Microsoft, and VeriSign. Their original specification was published on 5 April 2002, and was followed up by an addendum on 18 August 2002. WS-Security addresses security by leveraging existing standards and specification. This avoids the necessity to define a complete security solution within WS-Security the industry has solved many of these problems, Kerberos and X.509 address authentication. X.509 also uses existing PKI for key management. XML Encryption and XML Signature describe ways of encrypting and signing the contents of XML messages. XML canonicalization describes ways of making the XML ready to be signed and encrypted. What WS-Security adds to existing specifications is a framework to embed these mechanisms into a SOAP message. This is done in a transport-neutral fashion. OASIS proposed "The Web Services extensible Access Control Language (WS-XACML)" as XML based language to specify and exchange access control policies. WS-Security is designed to define authorization policies for principals that are specified using XML.

WS-XACML assertion matching:

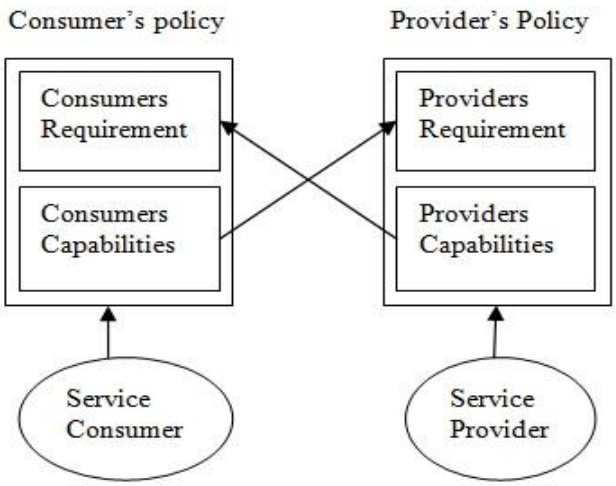

Fig 2: WS-Security assertion matching

The major problem with the admitted solutions for the enforcement of web services security is the static embedding of the features for the enforcement of web services security is the static embedding of the security features in the code/design of the web services. Many security features require run-time verification of the security policies, which may often be modified and updated. This means that when the security policies and/or the verification strategy change, the developer has to go back to the design/code of the web services and update them accordingly. This mechanism is bulky, error-prone and tedious. Our approach relies on the dynamic injection of AOP aspects into BEPL processes. This allows to easily update the security measures when needed, without affecting the business logic of the BPEL process.

Regarding the use of AOP security the following is a brief overview of the available contributions. Digital labs proposed an AOP language called CSAW, which is a small superset of Ph.D. thesis, discussed an aspect-oriented approach that allowed the integration of security aspect within applications. It is based on AOSD concepts to specify the behavior code to be merged in the application and the location where this code should be injected. The approaches in the AOP are useful to explore the feasibility of using AOP in software security. Hence, we can benefit from their achievements in building our security model.

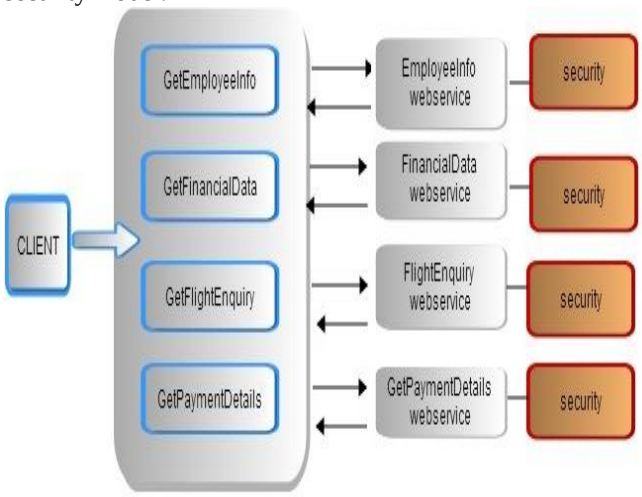

Fig 3: Architecture of Flight System(FS)

Fig. 3 explores the interaction between the user, the BPEL process and the web services of the flight system. As depicted in the fig, the security features are deployed on the web services side (i.e. not in the BPEL process). This clearly shows that any changes in these security features need a modification in the corresponding web service.

Our flight system is mainly composed of four separate web services, a BPEL process and a graphical user interface that allows the user to request information about the flight agency staff, get available flights and its monthly revenues, and make a reservation. The system available services are shown in the system main page. First, the financial data service allows the user to request the revenues and expenses of the flight agency for a given month, second, the flight inquiry service returns a list of the airline, and the available seats and tickets price. The employee information service allows the user to view information about the flight system staff by entering their ID number. This information includes the employees full name, phone number, email, address, post and his office number. Finally, the make reservation service enables the user to reserve a seat on a certain flight. All users who can access the system have records in the database. In other words, each user has an ID and a password stored in the database, in addition to his/her personal information. Each time a user wishes to access one of the flight system services, both the authentication and access control services are invoked to ensure that he/she is not only a valid user, but he/she also has the permission to view the requested information.

III. APPROACH DESCRIPTION

Aspect Oriented Programming (AOP) is one of the most prominent paradigms that have been devised for integrating non-functional requirements (e.g. security) into software. The main objective of AOP is to have a separation between cross-cutting concerns. This is achieved through the definition of aspects. Each aspect is a separate module in which point cuts are defined. A point cut identifies one or more join points. A join point identifies one or many flow points in a program (in our case a program is a BPEL process). At these points, some advices will be executed. An advice contains some code that can alter the process behavior at a certain flow point. The integration of aspect within the application code is called weaving and is performed through one of the weaving technologies (e.g. AspectJ).

Security is one of the software aspect that are important to deal with, generally, developers do not separate between security and business logic codes. This means that any change in the security strategy has to be done on the application code, which can have impact on the business logic. AOP solves this issue by embedding security in aspects. Aspects allow to application, which make them interesting solutions for many security issues. Many contributions in addition to our experiments, have proven the usefulness of AOP for integrating security features into software.

In this context, we present in this section an aspect-oriented approach for the dynamic enforcement of web services security. Our proposition is based on the use of AOP in the BPEL process of the composed Web services. It allows to specify the security concerns into separate components called aspects. These aspects are then weaved in the BPEL process at runtime.

IV. CONCLUSION

We presented an approach to use language based information flow control to ensure the confidentially, integrity of user's data provided to dynamically composed web services. Our hypothesis is based on the coactions between AOP and formation of web services. It permits the partition of business and security concerns of web services, and hence building them independently. It also permits the alteration of the web services at run time and provides distinction for designing cross-cutting concerns between web services.

REFERENCES

1. Benslimane, D.; Dustdar, S.; Sheth, A. (2008). "Services Mashups: The New Generation of Web Applications". IEEE Internet Computing 10 (5): 13–15.doi:10.1109/MIC.2008.110
2. Maler, Eve. "Minutes of 9 January 2001 Security Services TC telecon". security-services at oasis-open mailing list. Retrieved 7 April 2011.
3. Bob Atkinson, et. al.: Web Services Security (WS-Security) http://www.oasis-open.org/committees/tc home.php?wg abbrev=wss.
4. Anne Anderson, "WS-XACML:Authorization and Privacy Policies for Web Services" https://www.oasis-open.org
5. F. Paci, E. Bertino, and J. Crampton, "An Access-Control Framework for WS-BPEL," International Journal of Web Services Research, vol. 5, no. 3, pp. 20–43, 2008.