ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 3 (July - Sept. 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

An Architecture for Securing Heterogeneous Web Services

[I]G. Bharathy, [II]P. Joseph Charles, [III]Dr. S. Britto Ramesh Kumar
[I]M.Phil Scholar, [II]Assistant Professor, [III]Research Advisor
[I,III]Dept. of computer science, St.Joseph's college, Trichy, Tamilnadu, India.
[II]Dept. of Information Technology, St.Joseph's college, Trichy, Tamilnadu, India.

Abstract
The conventional access control mechanisms are based on the preliminary identification and authentication of the access requester. The server does not have enough knowledge about the service requesters in order to allow for the open Web service systems. The aim is to collect the information about the service requester in the initial process and verify the details whenever the service requester request for the services. The access control based on the attributes value will confirm the authorized user. The Attribute based Access Control create a policy using the attributes of participant entity. The request of Web service may access these operations using the standard XML message and provide the implement method for SOA. XACML adequately represent the attribute based access control policy, Policy Decision Point (PDP) implement authorization policy depending on the attribute information of body, object and environment.

Keywords
Security, ABAC, XML, Heterogeneous Web Service

I. Introduction

Web Services represents a new architectural paradigm for web application. Web services implement capabilities that are available to other applications via industry standard network and application interface and protocols. Access control policy is used to protecting resource not to be accessed by the illegal user, so as to specify the legal user operating resource, the access control policy determine which certificates are announced, and by which order are announced.

An application can use the capabilities of Web Services by simply invoking it across a network with having to integrate it. Service-oriented architecture (SOA) is a component model. The different functional units of applications (called services) were linked through well-defined interfaces and contracts. It is an independent of the implementation services, hardware platforms, operating systems and programming languages. Web services essentially involve the three roles of Service Oriented Architecture (SOA), are service broker, service requestor and service provider.

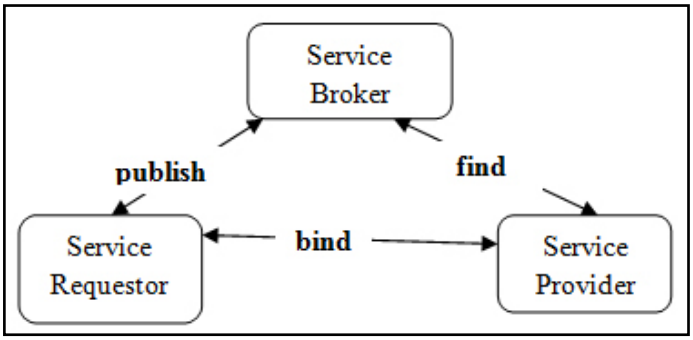

Fig.1.1: Service Oriented Architecture

A. Web Services Standards and Technologies

XML (eXtensible Markup Language) is the fundamental technology that underpins everything else in Web services. Of paramount importance to the XML suite of technologies is XML Schema, which provides a meta-level description of XML content. XML Schema can, in the simplest sense, be thought of as a means of dictating the format and content of XML documents. However, XML Schema's real power lies in the fact it can be used as a platform independent type description language, where XML documents are then used to transport data in accordance with those type descriptions

Web services are software components that expose their functionality to the network. To exploit that functionality, Web service consumers must be able to bind to a service and invoke its operations via its interface. To support this, two protocols that are the fundamental building blocks on which all else in the Web services are predicated: SOAP and WSDL.

A SOAP (Simple Object Access Protocol) message is an XML document whose root element is called the envelope. Within the envelope, there are two child elements called the header and the body. The Web Service Description Language or WSDL is used to describe a Web services endpoints to other software agents with which it will interact. WSDL can be used to specify the interfaces of Web services bound to a number of protocols including HTTP, GET and POST, since it is SOAP which consider supporting the (logical) Web services network. The UDDI (Universal Description Discovery and Integration) is a registry and a protocol for publishing and discovering Web services.

B. Attribute Based Access Control

ABAC is one of the frequently used access control model which is used to implement access control policy. Attributes are the entity of the body and object. It is used to find the authorization by determining the attributes of the request body, object and system. It gives Web service access control having stranger express, access control policy depend on the request body attribute (such as identity, address and so on). XACML (eXtensible Access Control Markup Language) adequately embody the course of attribute based access control policy, PDP (Policy Decision Point) implement authorization policy depending on the attribute information of body, object and circumstance. ABAC fit eh distributed and heterogeneous environment and it provide the functions of attribute defining and attribute certificate maintaining.

C. Security Limitations in Web Services

Security has become the key issue in the field of web services technology. As a result, Web service security needs to have more concern for the major security issues such as authentication, authorization, confidentiality, integrity and non-repudiation.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

II. Related works

Access control is the component of security systems responsible to evaluate if a subject can be allowed to operate in a given way on a specific resource. Several access control models are based on logic expressions, and prescribe access decisions on the basis of some properties of the requesting party, which can be proven by presenting one or more certificates. All these solutions are typically based on logic-based access control languages, which are powerful, highly expressive, and permit to specify recursive conditions and complex relations between parties in a simple yet effective way.

Kuyoro et al. [1] focused on the general framework of security issues and the proposed solution to web services security risks. Web Services provides interoperability across security policy domains. The threats are shared with conventional web application system, while others are specific to web services. The general security threats that can occur in any web application are listed as follows, SQL Infections, Capture and Replay Attacks, Denial-of-Service Attacks, Improper Error Handling, Eavesdropping and Session Hijacking.

The XML signature and XML encryption specification provide standard methods for digitally signing and encrypting XML documents including SOAP messages. This provides persistent confidentiality beyond a single SOAP communication. The SSL (Secure Socket Layer) creates a secure tunnel in between originator and destination computers based on public key encryption technique. The author doesn't mention any new security architecture based upon web services that support authentication, authorization and integrity.

Song Guo et al. [2] was described that the Web service provider can confirm the Web service request, verify the identity and authorization of the request, and ensure to provide the security Web service. Attribute based automated trust negotiation stand for a new the security technology, which provide the safeguard for realizing the span-domain resource sharing and accessing, have the single time exchanging the trust certificate, the less network expense, the less storing the certificates, the more effective preventing the middle attack, the higher security in comparison with the traditional automated trust negotiation.

IBAC (Identity based access control, IBAC), use the matrix based access control, associate authority with body identity. At present, the resources researched which involve the sensitive information have two sorts, the resource content is sensitive.

Hua Yue and Xu Tao [3] suggest the frame work for the web services security problem. With the development and universal application of SOA technology, security issues of Web services based on heterogeneous platform have become increasingly prominent. Web services provide a range of open standards-based solution to both business-level and application-level interoperability architectures. The framework can achieve a separation of the safe logic and the business logic. By combining a series of platform-neutral technology, we can implement internet and intranet services and applications on the fast delivering and using. It can provide a new way in order to take full advantage of various forms of network resources.

C.A. Ardagna et al. [4] illustrate a model that can be deployed in the XACML standard by exploiting its extension points for the definition of new functions, and introducing a dialog management framework to enable access control interactions between Web service clients and servers. XACML also permits expressing the fact that some properties should be certified; this is limited however to conditions on attributes issuer, time, and date that can be associated with the property. While XACML acknowledges that properties can be presented by means of certificates, and as a matter of fact, it has been designed to be integrated with the Security Assertion Markup Language (SAML) for exchanging various types of security assertions. The project is committed to building a robust prototype of a system supporting the management of privacy policies.

Kuyoro et al. [5] present that the QOS-Quality of Services is usually employed for describing the non-functional characteristics of Web services and employed as an important differentiating point of different Web services. Web services are designed to be accessed by other applications and vary in complexity from simple operations, such as checking a banking account balance online, to complex processes running CRM and ERP. Web services are hardware, programming language and operating system independent. QOS–aware offers in order to gain the highest possible profit from their business. The drawback is that the providers need to specify and guarantee the QOS in their web services to remain competitive and achieve the highest possible revenue their business.

III. Proposed Architecture

This chapter presents and discusses the security architecture developed for the credentials authentication, authorizations, access control, etc between the Service Requestor and Service Provider.

It explains the secure transmission of the web services between the client and the server with the security entities (attributes) which is given by the user in the registration process. The security for the proposed architecture provides the software elements that make up web service security. These elements perform various kinds of security tasks. The security architecture elements are cohesive and loosely coupled to simplify reuse and maintains. The client-side User Interface (UI) elements provide a mechanism for service requestor to interact with application deployed in the web server. These elements acquire and validate data entered by the user. UI process elements drive security processes to provide secure user interface for authentication, authorization, confidentiality, accountability, non-repudiation, etc. and offer communication security with the help of Security Entities. The Security Entities manage the user credentials such as client and server certificates. The Data Access Elements are used to access those security credentials that reside at the client.

The repository consists of the database information about the security, service provider and the service requestor. The Service Agent contains the list of services, security information and the database. The service Agents presents the unique feature of request façade that is used to optimize the user's requests. Security Logic elements keep the security logic for the accessing the services. These elements apply security rules on interfaces and perform various security services. The Security Entities manage all the user's credentials and implement the security properties after the request process. The Data Access Elements abstract the logic required to access the underlying services which is in data store. The Service Interface Elements exposes the business logic as well as security logic into the all web services.

The Security Manager collects the Attributes exposes the business logic as well as security logic into the all the web services. Security Workflow elements of the service agent establish and maintain the security services while accessing the service requests. The

Information Access Elements provides the secure way to access all the services available in the Service list. The Information Access Elements and Data Access Elements in the service requestor are used to receive the information about the user and maintain the backup. The Security Agent also contains the Information Access Elements and Data Access Elements. In this, the Service Agent receives the information from the service requester and allows it to store in the Repository.

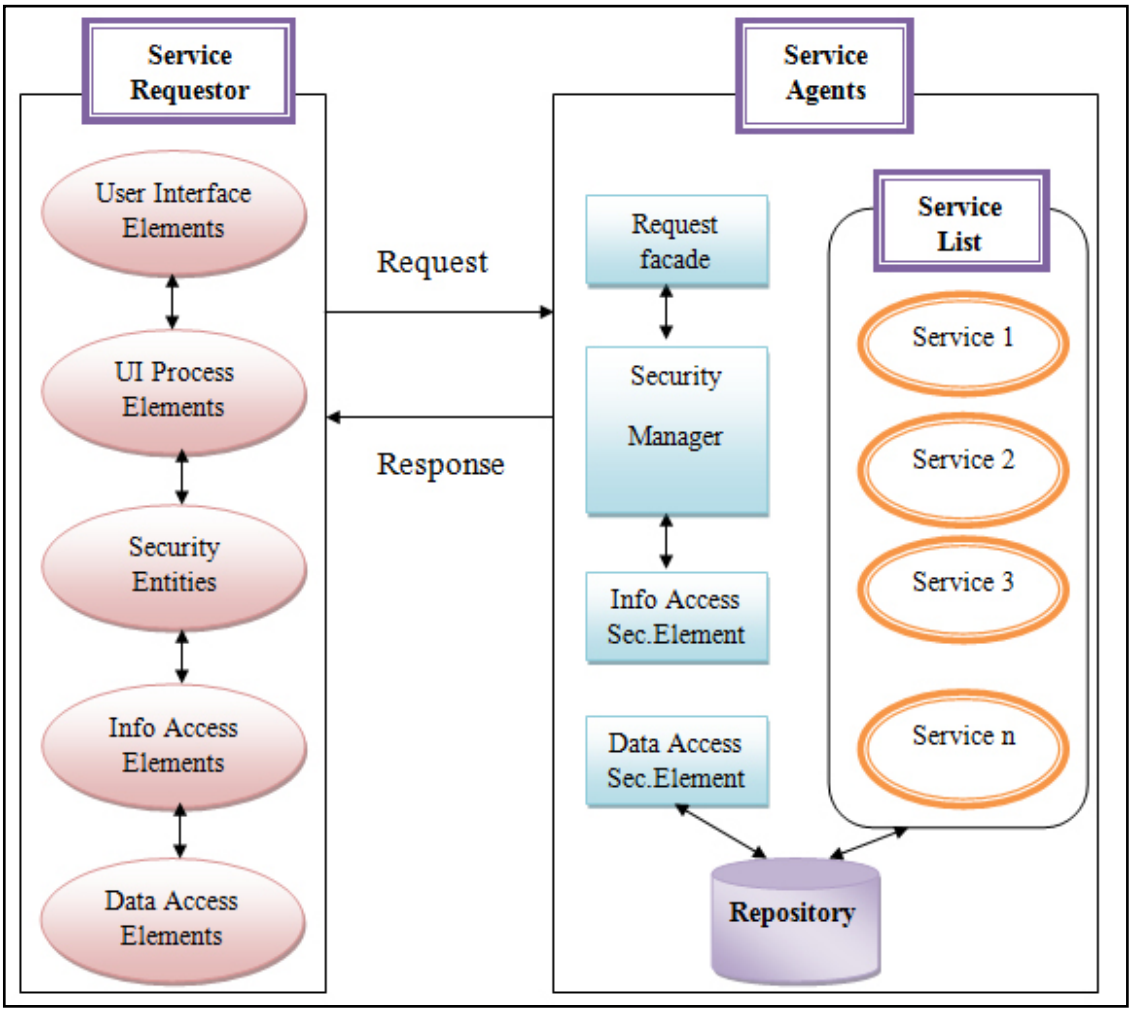

Fig. 3.1: Proposed architecture for securing Heterogeneous Web Services

The proposed security architecture offers security features such as authentication, authorization, confidentiality, data integrity and non-repudiation, and resolves frauds user to access the web services. The authentication is a process of verifying authenticity of the user to access a system by one or more authentication mechanisms. This system supports the entity authentication and user authentication. Authorization is the property by which the user's properties can be associated with the attribute to access.
Confidentiality is the process that keeps the information secret from an unauthorized person, process and device. Integrity ensures that the information and systems have not been altered or corrupted by outside parties by the security entities. Finally, the important property of non-repudiation prevents the communicating parties from denying their actions after the transaction has completed by issuing proof.
The security manager will check the five attributes and allow the service requestor to access the available web services. When the request arises in the service agent, it will check the availability of the service.
The XACML policy a conjunction of five conditions on attributes nationality, city-birth, education profile, working class and year-birth, respectively that, according to the value of attribute Disclosure of the corresponding elements. Apply, associated with disclosure policies property, condition, and predicate, respectively. A disclosure policy that specifies what information in a condition cannot be released is then enforced by hiding such Information. To this purpose, a special keyword undisclosed will be used in the XACML policy to indicate information hidden to comply with the selected disclosure policy. Keyword undisclosed is equivalent to the empty square brackets in our formal model. For instance, consider again the XACML policy. For instance id document can be defined as an abstraction for any element in the set of credentials {identity card, driver license, passport}. An authorization specifying that the requester needs to provide an id document for accessing a specific Web service can then be satisfied by the requester presenting any of the three credentials above. XACML does not provide explicit support for abstractions.
Based on logic, such approaches, while appealing for the expressiveness, result not applicable in practice, where simplicity, efficiency, and consistency with consolidated technology are crucial. The eXtensible Access Control Markup Language (XACML) has established itself as the emerging technological solution for controlling access in an interoperable and flexible way. Although supporting the most common policy representation

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

mechanisms and having acquired a significant spread in the research community and the industry, XACML still suffers from some limitations which impact its ability to support actual requirements of open Web-based systems. The paradigm shift from requester identification to access condition communication and the need for a general system behavior description has already been acknowledged by the research community.

Many works and progresses in credential-based and attribute-based access control rely on the idea that the server communicates to the requester the credentials that it must possess, or the properties that needs to satisfy, to acquire access. Several works have also investigated the different aspects of credential-based access control, and presented different models and languages.

Typically, a logic-based language is proposed, allowing for compact and expressive specifications of the access control policy as well as its communication to the requester. Furthermore, when privacy is an issue, these works assume that the requester can enforce her own policy and initiate a negotiation with the server, during which access policies, credentials, and attributes are exchanged, until access is eventually granted or denied

The access control languages for an open world scenario, which provide flexibility and interoperability. eXtensible Access Control Markup Language (XACML) is an XML-based language for expressing and interchanging access control policies. XACML defines both a modular and distributed architecture for the evaluation of policies, and a communication protocol for message interchange. Each access request is received by the Policy Enforcement Point (PEP) that is responsible for the enforcement of each access decision. The Policy Decision Point (PDP) is the component that produces the access decision for each access request by retrieving the applicable policies from the Policy Administration Point (PAP) that provides functionalities to administer access control policies. The PDP must receive all information relevant for the decision process.

The PIP interacts with the Subject, Resource, and Environment modules. The Environment module provides a set of attributes that are relevant to take an access decision and are independent from a particular subject, resource, or action. The PEP fulfills the obligations and, if permitted, it gives access to the requester. Otherwise, access is denied. The specific focus of the project is toward the definition of a novel policy language and architecture able to support the privacy requirements of individuals interested in exploiting the services of the Web, keeping control on their data.

IV. Conclusion

The focus is toward the definition of a novel policy language and architecture able to support the privacy requirements of individuals interested in exploiting the services of the Web, keeping control on their data. In this scenario, XACML has been identified as a promising solution thanks to its technical features and the widespread support. This architecture is committed to building a robust prototype of a system supporting the management of privacy policies. To overcoming this problem some policies are created on the basis of the services available and that policies are taken in account during the services requestor login. It valid the user preferences and use the identity to be part of the key to the actual data.

V. Future Enhancement

The proposed architecture for the Securing Heterogeneous Web Services is implemented with some policies which are developed using the XML. The XACML implementation is also being tested and is expected to see better results than the using policy. It also makes way for the ubiquitous environments to be safe in controlling access to any kind of sensitive information. In the future work, this XACML policy can propose in the mobile environment and a mechanism can also develop to maintain the trust and the security.

References

[1] *Kuyoro Shade O, Ibikunle Frank, Awodele O and Okolie Samuel O, "Security Issues in Web Services", IJCSNS International Journal of Computer Science and Network Security, Vol .12 No.1, January 2012.*

[2] *Kuyoro Shade O, Ibikunle Frank, Awodele O and Okolie Samuel O, "Quality of Service (QOS) Issues in Web Services", IJCSNS International Journal of Computer Science and Network Security, Vol .12 No.1, January 2012.*

[3] *Song GUO and Xiaping LAI, "An Access Control Approach of Multi_Security Domain for Web Service", Advanced in Control Engineering and Information Science, Procedia Engineering 15(2011) 3376-3382.*

[4] *Hua Yue and Xu Tao,"Web Services Security Problem in Service-Oriented Architecture", 2012 International Conference on Applied Physics and Industrial Engineering, Physics Procedia 24(2012) 1635-1641.*

[5] *C.A. Ardagna, S. De Capitani di Vimercati, S. Paraboschi, E. Pedrini, P. Samarati and M. Verdicchio, " Expressive and Deployable Access Control in Open Web Service Applications", IEEE 2010, Department of Information Technology, Universit a degli Studi di Milano, 26013 Crema, Italy, pp1-14.*

[6] *S. Arunkumar, A. Raghavendra, D Weerasinghe, D Patel and M Rajarajan, " Policy extension for data access control", IEEE 2010, School of Engineering and Mathematical Sciences, City University London, Northampton Square, London EC1V 0HB, pp55-60.*

[7] *Xiaowei Li and Yuan Xue," A Survey on Web Application Security", Department of Electrical Engineering and Computer Science, Vanderbilt University.*

[8] *Kou Hongzhao, "A Study on the Security Mechanism for Web Services", World Congress on Engineering and Computer Science 2010 Vol 1, WCECS 2010, October 20-22, 2010, San Francisco, USA*