A TRUST BASED ACCESS CONTROL IN WEB SERVICES

Shamila .E.S[1], Ramachandran .V[2]
[1]Sathyabama University, [2]Anna University, Chennai
E-mail : [1]shamiiee@rediffmail.com

Abstract

Securing the web services is an issue has always been with the development of the Internet. There are much more serious security issues in a web services due to its openness and autonomy. . Building a trust mechanism in web services environment is very important. This paper presented a trust model of choosing trusted source Web Services. Our system collects the trust evidence of the clients, defines the trust polices, builds up the trust levels of the clients based on the trust evidence and policies. As more evidence becomes available, the system iteratively updates the trust information including trust evidence and polices. This paper also focus on using evidence for making access control decisions in web services environments. Our approach views access control as the filtering of messages between communicating services. We implement our evidence-based approach with a mechanism analogous to a network firewall, filtering messages going to and from a service.

Keywords: Web Services, Access Control, Trust, Firewall, Evidence, Security.

I. INTRODUCTION

Web services provide a standard framework for interoperating independently developed Web applications. Generally speaking, a Web service is a set of related functionalities that can be programmatically accessed through the Web. These functionalities represent the different operations made available by the Web service and are described in its service description using the WSDL standard language.

A. Access control

The goal of access control is to allow only authorized users to access sensitive information. Access control is emerging as a generalized approach to security and has been shown to be applicable to a wide range of security requirements of organizations and applications. Possibility of using Access Control approach to an environment with multiple policy domains further justifies the tremendous momentum seen in Access Control research in the recent years. However, with the increasing size of the problem domain represented by today's huge information systems that cross organizational boundaries, the issue of management and complexity of security and providing its assurance poses daunting challenges. A fuzzy approach to addressing the issue of security can provide a pragmatic and promising new direction in this area.

A.a. An Access Control model.

Access control denotes the fact of determining whether a user (not necessarily an human user: process, computer, etc.) is able to perform an operation (read, write, execute, delete, search, etc.) on an object (more generally: a tuple in a database, a table, an object, a file, etc.). An operation right on an object is called permission. An access control model define how to organize the permissions of users.

Authorization

Process of granting or denying access rights for a resource to an authenticated user. (What are you allowed to do?)

Credentials

Information that includes identification and proof of identification that is used to gain access to resources. Examples of credentials are user names and passwords, smart cards, and certificates.

B. Security

Security access control mechanisms play a key role in the overall structure of any security system. They are responsible for controlling the access permissions to system resources; i.e. determining who has access to which resource and with what type of access. Access control mechanisms rely on the authentication mechanisms to identify the users and ensuring that they are actually who they claim to be. The most common authentication method used to date is the user ID and password (or PIN number) combination, though other methods, such bio-metric identification, have been used with varying degrees of success. Security is the biggest challenge facing web services deployments today. Every customer and web service has its

own security requirements based upon their business needs and operational environment.

The security challenges presented by the Web services approach are formidable and unavoidable. Because a Web service relies on some of the same underlying HTTP and Web-based architecture as common Web applications, it is susceptible to similar threats and vulnerabilities. Many of the features that make Web services attractive, including greater accessibility to data,

dynamic application -to-application connections, and lack of a well -defined and bounded run-time environment, are at odds with traditional security approaches

C. Trust in Web Services

Security and Trust are two important concepts in information security. Trust is usually based on people's experiences. Trust is an expectation to a component's some special behaviors with some prerequisites, when a set of prerequisites are available, then it can assure that some particular behaviors will be performed as expected.

With the rapid development of Web Services systems recently, they attract increasing attention from researchers, but they also bring up some problems. Some web services might be buggy and cannot provide services as they advertise. Some might be malicious by providing bad services to get more benefit.

Trust is at the core of most relationships between human beings. Take the simple example of purchasing an item from a shop. We may choose to buy a certain brand because we have found it to be trustworthy in the past or it has a reputation for being widely “trusted”.

Trust between web services begins to mirror those real-world relationships in the society. Because of the importance of trust and reputation, the formalization and quantification of trust becomes increasingly important, some researches have been done in the field.

Now the fuzzy mathematics trust model only uses fuzzy set and subject degree definitions to represent trust concept, how to use deep fuzzy mathematics theory to compute trust needs further research. In this paper, we give a fuzzy relation trust model of choosing trusted source web service to enhance the trust degree.

As a consequence of the rapid growth of Web Services, the issue of trust becomes

central for businesses. There are no accepted techniques or tools for specification and reasoning about trust. There is a need for a high-level, abstract way of specifying and managing trust, which can be easily integrated into applications and used on any platform. A typical application requiring a formal trust decision becomes apparent when service consumers are faced with the inevitability of selecting the “right” service. The distributed nature of these services across multiple domains and organizations, not all of which may be trusted to the same extent, makes the decision of selecting the “right” service a demanding concern especially if the selection proves to be automated and performed by an intelligent agent. The above challenges necessitates the introduction of a new framework addressing these challenges and enabling users to:

- Describe and discover Web Services semantically.
- Focus on the conceptual basis of their experiments rather than understand the low level details of locating services.
- Automatically select a trustworthy service that meets the user's trust policy.

II. EVIDENCE - BASED ACCESS MODEL

In this section, we introduce the high-level goals of our design and provide an overview of the concepts.

The goals of our model are to:

1. Support communication between parties in different trust domains or in cases where no pre-existing trust domains exist. This goal requires that parties can name each other securely and that the model interoperates with existing access models.

2. Allow evidence associated with parties to be securely collected and evaluated for the purpose of allowing access to resources. In particular, access decisions should be based on policies that are capable of balancing security with convenience.

3. Create an ecosystem in which evidence providers can flourish. This goal requires a system for creating evidence, associating evidence with parties, and trading evidence between parties. New forms of evidence and new policies must be allowed to be introduced freely.

In our work, we separate unique identifiers from the information used to make decisions about access control. We call the identifiers entities to highlight the fact that they do not necessarily represent a user, but can uniquely name any available service. We assume that entities are uniquely defined and can provide cryptographic signatures that are sufficiently strong for the task they are being used for based on existing technology, such as a public key infrastructure.

With the traditional concept of user, identity and trust are tightly coupled. Users verify their identity by providing a password, after which, the user and the capabilities of the user are co-mingled. In our model there is no preconceived relationship between entities.

The relationship emerges locally at every service based on available evidence. While entities can be uniquely named and addressed, their names are difficult to remember, communicate, etc. To support a more convenient way to address entities and their related services, we use URIs (uniform resource identifiers), and we call an entity/URI pair a port. Messages are sent to and from ports.

A message identifies the port it comes from, the port it is being sent to, and contains the structured contents of the message. We assume that messages contain a distinguished field called action, which we interpret as the verb of the message.

In our model, a service is a container of state, with zero or more ports, that is the sender or receiver of a message, named by a port, whose behavior upon receiving a message depends on the message and the state of the service. Upon receiving a message, a service can change its state and/or send zero or more messages.

A resource is a part of a service's state that can be serialized and sent through a port in a message. A document is a resource that has been serialized, for example, into XML. We call the sender of a message requesting access to a resource a client.

A firewall is a processor, associated with a service, that processes all messages that are sent to or from the ports of the service. A service's firewall may access all the service's state.

For every message sent or received, the firewall implements the function f(message, state) that produces one of the following results:

1) the message, in whole or in part, is delivered to or from the port,

2) the message is discarded and an error message is returned,

3) the message is silently discarded. If an error message is returned, it may contain additional information, such as the port of an evidence provider.

Firewalls manage the information needed to make access control decisions and execute the policies that determine the decisions.

Evidence is any state in a service that its firewall accesses in making an access control decision. Policies are the procedures whose evaluation results in access control decisions. While one can make the case that there is a duality between policies and evidence, for simplicity we consider only cases where policies are procedural and fixed by the entity that a firewall protects and evidence is data that is accumulated on a per-requesting-service basis. An evidence provider is a service that interacts with the environment and other services and provides data that are pertinent to firewall decisions.

III. PROPOSED WORK

This paper considers the problem of providing access control using evidence based. One approach to access control is based on authenticating users. In this model, users are authenticated by providing a password to a central authority. Once they have established their identity, they cross the trust boundary and are granted credentials that then allow them to access resources.

By evidence, we mean any knowledge that relates to the requesting or responding service, potentially including reputation , recommendations, passwords, HIPs , CAPTCHAs , client puzzles, biometrics , proximity , web service rating , credit reports, and quizzes. Evidence can conceivably be gathered, accumulated, and evolved as necessary over time. If a service needs more evidence, it can challenge the requesting service directly (e.g., by requesting a password), or it can request evidence about the requesting service from another service, an evidence provider.

This model captures identity in the body of collected evidence about other services that each service gathers. Access decisions are based on evidence and policies held locally by each service. Each service can have different evidence for the same requesting service, and the local evidence can be thought of as a partial view of the identity of the requesting service.

In this paper, we define a model for access control between services that filters messages between services with an evidence-based firewall (EBF). The evidence-based firewall manages the interaction between services and regulates what services have access to what resources. Figure 2 illustrates our approach, showing two communicating services that each have an EBF processing messages sent between the services. Based on access control policies and known evidence about the sender and receiver of the message, the EBF determines whether the message is allowed to be sent or received. Access control policies allow the EBF to make access control decisions based on a variety of sources, including the sender, receiver, and contents of the message.

We base our framework on the firewall concept because it is well understood, widely used, and effective. Firewalls limit both incoming and outgoing messages, necessary to prevent access and maintain privacy in web services computing environments. Furthermore, firewalls are also used to filter data that we classify as boring or annoying.

Evidence-based access control can be used to achieve traditional centralized access control if desired. Each user would be an entity and all the resources protected by a domain would require the same evidence, for example, that a user knows a password. Nothing in our model prevents multiple services from synchronizing their policies and evidence if desired.

Beyond providing new evidence, evidence providers might also provide evidence management services, including caching, replicating, summarizing, and/or distilling it into higher order evidence.

There are many ways that an access control decision can be made based on the sender, receiver, the message itself, and available evidence. We present pseudocode for a simple access control algorithm in Figure 2, which illustrates an algorithm with separate "in" and "out" policies, and uses evidence associated with the sender while making "in" decisions, and uses evidence associated with the receiver while making "out" decisions. We use such a policy organization in the scenarios that follow. While this figure depicts a Boolean policy, nothing prevents us from providing a more complex policy that allows more subtle access distinctions.

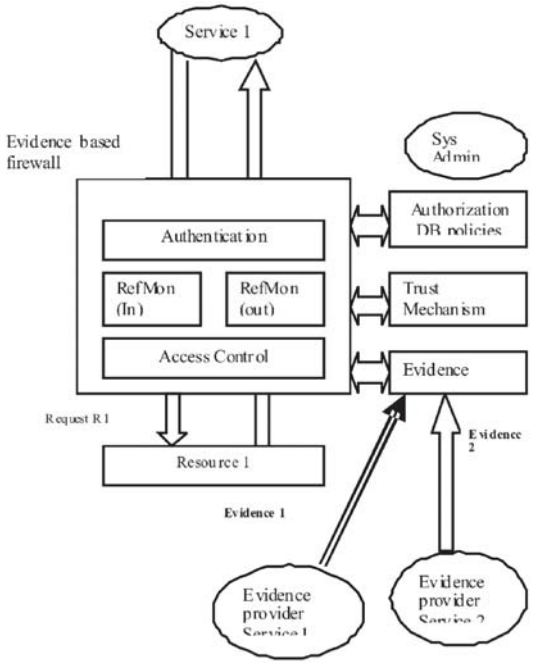

Fig. 1. Evidence Based Access Control

We assume two web service, the source web service and the target web service. The source web service request the service from the target web service, so the source web service can be treated as service customer while the target web service is service provider. The source web service may calculate the combined trust value about the target web service to decide whether to send the request information to the target web service. If the target web service received the request information, it should calculate the combined trust value about source web service to decide whether to provide service to it. The process of calculation of combined trust value is the same for each web service, but the source web service will give the evaluation to the exchange.

When a web service wants to know another web service's combined trust value, it will request a reference to the target web service from other web services. The target's combined trust value will be calculated based on two types of direct trust and reputation.

Web services can set up the proportion of two types of direct trust and reputation for themselves. If the calculated combined trust value is greater than the threshold value, an exchange will be performed. At the end of the exchange, the service customer will evaluate the service provider based on its service quality, and the direct experience trust and recommendation trust will be updated with the combination of trust values and the evaluation. If the exchange is satisfactory, the trust degree of the service provider will be increased. If the recommendation is consistent with the evaluation of the exchange, the web service will increase the recommendation trust degree of the recommendation provider, otherwise decrease the recommendation trust degree. If the recommendation trust value is lower than a threshold value, the recommender will be eliminated from the group of recommendation web services while some other web services will be added into the group.

```
boolean Access?(port sender, port receiver, action, body) {
boolean decision, error;
policy p;
evidence e;
if (sender == me) {
p = CheckkupPolicy(me, "out");
e = CheckupEvidence(receiver);
}else {
p = CheckkupPolicy(me, "in");
e = CheckkupEvidence(sender);
(decision, error) = p(e, action, body);
if (error) {
if (protocol_requires_response)
Send(receiver, sender, error);
return false;
return decision;
```

Fig. 2. Access Decision Procedure

V. RELATED WORK

There is a large body of work on the general issue of access control in a distributed environment, while less work focuses on the problems specific to web services. While some prior work has considered a general framework using evidence for access control, much of the previous work assumes a small, fixed set of evidence sources.

With our EBF approach, the main focus is on the evidence itself. With this focus, we hope to stimulate

research in ways to evolve, manage, combine, and generate new evidence.

VI. SUMMARY

We describe an evidence-based access model that assumes no centralized domain, focuses on access control to non-critical data, and encourages a robust ecosystem of new evidence and policy providers along with trust. Our model binds names to evidence locally, so that every service has a potentially different view of the credentials of client services. We enforce access control by interposing a firewall on every incoming and outgoing message, allowing message filtering based on the sender, receiver, and the contents of the message. And also trusting the web services based on threshold by both direct and indirect trust. Because much data is either dangerous or critical, relatively benign message filtering is an important part of any access control system.

VII. REFERENCES

[1] Lei Wang, Yanqin Zhu, Lanfang Jin, Xizhao Luo, Trust Mechanism in Distributed Access Control Model of P2P Networks, In Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science, 978-0-7695-3131-1/08 , 2008.

[2] Johannes Helander and Benjamin Zorn. Medina: Combining evidence to build trust. In Proceedings of the Workshop on Web 2.0 Security and Privacy 2007 (W2SP'07), May 2007.

[3] Yi Chen, Junzhou Luo, Xudong Ni, A Fuzzy Trust Evaluation Based Access Control in Grid Environment, , In Proceedings of the The Third ChinaGrid Annual Conference, 978-0-7695-3306-3/08, 2008.

[4] A Flexible Trust Model for Distributed Service Infrastructures, Zhaoyu Liu, Stephen S. Yau, Dichao Peng, Yin Yin, In Proceedings of the 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC), 978-0-7695-3132-8/08, 2008.

[5] Ali Shaikh Ali, Simone A. Ludwig, Omer F. Rana , A Cognitive Trust-Based Approach forWeb Service Discovery and Selection, In the proceedings of Third European Conference on Web Services (ECOWS'05) 0-7695-2484-2/05, 2005

Shamila .E.S, is working as senior lecturer in department of computer applications at sathyabama university. She is having around 10 years of experience in teaching. She has published papers in various conferences and journals. Her area of interest is web services security and trust management.