Enhance the XML Documents Security using Web Services

K.L.N. Durga, Dr. N. Chandra Sekhar Reddy² Chaitanya3
M.Tech CSE Dept., Institute of Aeronautical Engineering, HYD-500043, AP, India.
Professor, CSE Dept., Institute of Aeronautical Engineering, HYD-500043, AP, India.
Professor, CSE Dept., Institute of Aeronautical Engineering, HYD-500043, AP, India.

ABSTRACT:

An increasing amount of semi-structured data, XML has become fundamental to humans and programs. XML changed by the World Wide Web Consortium (W3C) is quickly emerging as a new standard language for semi-structured data representation and transfer on the Internet. XML documents usually hold private information that cannot be shared by all user communities. So securing XML data is becoming increasingly important and several approaches have been created to protect information in websites. Usage control has been considered as the next generation access control model with differentiating properties of decision continuity. It has been proven to be efficient to improve security administration with flexible authorization management. Usage control changes finer-grained control over usage of digital objects than that of traditional access control policies and models. XML schema defines format and contents of xml instance. Access control established on a schema will be transported to all its instances. Also, XML schema supplies a mechanism to build relationships between schemas and elements. We present a usage control model to protect information distributed on the web. As privacy becomes a major refer for consumers and enterprises, many research have been concentrated on the privacy protecting technology in recent years. In this paper, We present a comprehensive approach for usage access control established on the notion purpose. In our model, purpose information connected with a given data element specifies the intended use of the subjects and objects in the usage access control model. A key feature of our model is that it accepts when an access is required, the access purpose is checked against the intended purposes for the data item. We suggest an approach to represent purpose information to support access control established on purpose information. Finally comparisons with related works are analyzed.

KEYWORDS: XML Schema, Access control policies, XML Signature, Fine grained approach.

1. INTRODUCTION:

The extensible markup language (XML) [1] is a standard for describing the structure of information and content on the Internet over the past several years. XML is a significant component in many XML web services and it is applied to store and modify data in the Internet environment that may include private messages of customers. It overcomes the complexity of standard Generalized Markup Language (SGML) and the user can specify document structures, removing the limit of the fixed tags in Hypertext Markup Language (HTML). XML schema is an XML-based alternative to DTD. It specifies the contents and relationships of elements in an xml instance. It supports complex constraints for XML components, such as elements, attributes, data types, and groups. XML schemas to define the format of information to be modified. With the features of XML schema, a flexible and easy-customized access control model can be achieved. Access control has been considered as a major issue in information[12] security community since the beginning of the information security discipline. Through access control, the system can limit unauthorized users access to the resources in the system and guarantees the confidentiality and integrity of the resources. Traditional access control models primarily consider static authorization decisions established on the subjects permissions on target objects. It focuses on the protection of data in a closed environment. More recently research in authorization is about trust management. Trust management relates authorization to a user's capability and properties. These access models have applied on the control of access to server-side objects. Our previous work has, however, some limitations. The first is that it has been developed
based on xml and object-relational data model with using usage access control[2]. But to manage purpose information that are complex, have hierarchical structures and are characterized by several semantic relationships, we need to develop a sophisticated purpose management model. The second is that does not adequately address the problem of how to find the purpose for which certain data are accessed by a given user. We believe that this issue may be satisfactorily addressed by believe on the usage access control model. However, in order to support qualifying for which purpose a certain can be accessed by a given object, we need to develop an extend usage model which is based on the purpose of subjects and objects. In this paper we introduce fine-grained method[11], an ever increasing amount of information is being made available in unstructured and semi structured form via web sites both on corporate Intranets and on the global Internet. The Internet should meet precise security requirements such as fine-grained authenticity, secrecy, non repudiation, and access control, involving data units at the level of granularity stipulated by communicating parties.

2. RELATED WORK:

Role Based Access Control (RBAC) mechanism for xml security[2],[4]. In their mechanism, privileges or permissions are defined based on xml schema components and will be transported to all instances, the permission reutilize through these hierarchies supplies the security[4]administration. Several constraints are presented in the mechanism the proposed mechanism can be modularly deployed and flexibly administrated in distributed environments the mechanism can be readily used to no schema established xml instances and instance level authorizations. However, our work substantially differs from that proposal. The main differences in our approach are in the following aspects. Firstly, their protocol is established on RBAC[6] and hence it focuses on permission-role assignment, object hierarchies and constraints. Our approach is established on usage control, we have analysed the characteristics of various access authorizations and presented detailed models for distinct kinds of authorizations. Secondly, their approach does not mention how to update user’s permission on xml [14] object when their conditions or obligations have changed. it is an important state for xml documents in the internet since users always alter their conditions. By contrast, users in our scheme have to pass pre-Authorizations and ongoing-Authorizations as well as pre- obligations, pre-conditions and ongoing-obligations and ongoing-conditions. This indicates that our mechanism is much more powerful in dynamic environments.

The Access control system supporting selective distribution of xml documents among possible larger user communities by using a range of key distribution methods[11]. In their papers, a formal model of access control policies[9,13] for xml documents is given it concentrates on key distribution methods to protect xml documents. The approach consists of encrypting distinct portions of the equal document according to distinct encryption keys, and selectively distributing these keys to the different users. By contrast, our work provides a rich variety of options that can deal with xml documents. Users can access xml documents with their properties are updated. In our scheme, users have to satisfy[14]. The access control system supporting [12] XML signature is a w3c recommendation for securing integrity of xml data during signing and/or verification process. An integrity function calculates digests of all desired xml fragments and computes digital signature of element signed info involving separate digest values DS of all desired xml fragments together with other auxiliary information needed by the xml signature recommendation. The root elements signature holds information about the signed set of xml fragments. A signature value and elements key information containing a key. That was used to sign element signed info. Element signed info further involves element canonicalization method that specifies an algorithm applied for normalization of element signed info before it is actually signed by an algorithm started in element Signature Method. Each element Reference within element Signed Info describes one digested xml fragment element Digest Method specifies a method applied for digesting the xml fragment, element Digest Value contains the digest value and finally, element transforms can hold arbitrary number of elements Transform and shows a
chain of transformations that are applied to prepare the xml fragment for digesting. The access control system supporting for xml injection is a collection of attacks, where a new part xml fragment is skillfully accepted to the current xml fragment. Where an xml fragment used by the integrity function is retrieved with a new xml fragment and the primitive one is protected to other location. Further, during the evolution of the integrity function, the primitive xml fragment is adjusted; however in sequential processing of the xml document the tampered newly enhanced xml fragment is employed. An access control system supporting xml injection is a collection of attacks[8], where a new part xml fragment is skillfully accepted to the current xml fragment. Where an xml fragment used by the integrity function is retrieved with a new xml fragment and the primitive one is protected to other location. Further, during the evolution of the integrity function, the primitive xml fragment is adjusted, however in sequential processing of the xml document the tampered newly enhanced xml fragment is employed. Fine grained access [11]control systems further excusing differential access rights to a group of using and allow flexibility in specifying the access rights of different users certain techniques are known for implementing fine grained access control. The fact that this employs a trusted server that acts the data in clear. Access control relies so software appears to arrange that a user can access a piece of data only if he is authorized to do so. This situation is not particularly appealing from a security stand point.

Obviously, access control in web services[10]will be popular in the future all messages and protocols in web services are in xml format. The schema based usage control model can be a solution of the web service security.

3. Existing System:

Web service represent a new and unexplored set of security-sensitive technologies that have been widely deployed by large companies, governments, financial institutions and in consumer applications. Unfortunately, the attributes that make web services attractive[8], such as their ease of use platform independence, use of HTTP and powerful functionality also make them a great target for attack and the web application attacks (SQL injection, XSS, etc…) can be reorganized to work with the next-generation of enterprise applications.

Attacks ON THE Web Services:

This section of this study specifies the different prominent attacks specific to the web services with their few proposed mitigation methods. Several attacks are demonstrated in a tree form.

Injection attacks: injection attacks generally deals with the attacks domain where malicious data is purposely included to input supplied to the web service to interrupt the normal functioning of web services. Some injection attacks are:

SQL Injection: it is an attack that generally occurs with databases[8],[5]. In this input query supplied to the database is changed in such a way to alter the output of the query as per the attacker requirement. Suppose, a web service is querying the database for the authentication of the user[7], the user supplies the input credential to the form and from the form the web service takes the credential for validation. In formal cases the validation is done and access is allowed to user but when the input is altered by attacker then unauthorized access can be derived by the user.

XML injection: this attack tries to change the xml structure of a SOAP message[3] by inserting content e.g. operation parameters containing xml tags. Like <author> and </author>. This can lead to xml injection attacks[10],[8].

Denial of service: it is the attack which targets the server behind the web service and derived at the request processing capability of the processor. In this attack the attacker manipulates the variables related with the request made to web service. Several types of DOS[10] attacks are there. Which includes xml and xml with http i.e. SOAP.

Oversize payload: this attack directs at exhausting the resources of the web server. Against web services[8], this is attack is easy to mount due to high
computation cost of xml processing. The reason behind this is that most web service frame works make use of tree based xml processing model like DOM (Document Object Model)[15]. Using this model xml document like SOAP request is totally read, parsed and changed into in-memory object representations which in case of large SOAP request take more computation power and memory than the original SOAP request.

**Malicious content:** An attacker sends a malicious code and attachment such as image[7], exe files and other application specific document with the SOAP message which holds malicious code, virus like Trojan horse and other which are communicated with the xml doc as the firewall which are designed to protect system against malicious code are blind to xml files. The SOAP with attachment specifications permits transmitting packages using MIME.

**Parameter tampering attack:** it is a kind of xml injection[10] attack in which the attacker changes the parameter s in a SOAP message in try to bypass the input validation in order to access the unauthorized information.

**Buffer overflow attack:** generally the communication between the sender and receiver of the web service is of asynchronous nature[8],[7], hence an attacker may send a malicious request containing specially fashioned parameters to overload the input buffer. Buffer overflows generally lead to crashes.

**Instantiation flooding:** As a new message gets a new instance of the BPEL process is generated. Such instance immediately starts executing the instructions given in the process description. Once all the instructions are executed only then the instance is terminated.

**Indirect flooding:** this attack works in equal way as the instantiation flooding attack but it directs the backend architecture. The attacker will flood the backend system with web service request messages and for each message a process instance is created.

**WSDL Scanning:** WSDL stands for Web Service Description Language and wsdl document associated to a particular web service contains all the details about the web service like endpoints, operations and parameters details, and binding details. WSDL documents are easily available on the web and an attacker can study the document to learn in depth about the target web service.

**Figure 1. The Taxonomy on the Attacks on Web services**

4. **PROPOSED SYSTEM MODEL:**

A new system for web attack detection, prevention and mitigation are presented[8]. It follows the anomaly-based approach, therefore known and unknown attacks can be detected. The system relies on a xml file to separate the incoming requests as normal or anomalous. The xml file, which is built from only normal traffic, holds a description of the normal behavior of the target web application statistically characterized. Any request which deviates from the normal behavior is considered an attack. The system has been used to protect a real web application.

**Counter Attack Techniques:** Generally used approaches to counter attacks are:

1. **Static Code Analysis:** “white-box” approach that consists of the analysis of the web application source code[8].
2. **Penetration Testing:** “black-box” approach that consists of the analysis of the web application execution in search for vulnerabilities. In this approach, the tester uses fuzzing techniques over the web http requests. Methods mostly applied to prevent an attack to occur, as shown in figure.
3. **Input Filtration:** Input taken from users must be filtered to remove escape characters
1. like ",",/,_ <script></script> and then only it is sent to web services for further processing.

4. **Centralized Schema Validation:**
   
   [8] Centralizing the validation process of request and response message within the network parameter with largely decreases the possibility of outside attack to a considerable amount.


6. **External References Blocking:** Blocking the external referenced element and using only[8] validated and cleaned urls can successfully avert the references attack.

7. **Schema Caching:** A secure repository of xml schemas that is periodically updated is one of the best ways to deal the usage of compromised, unapproved xml schema.

8. **Use of SAML for Authentication:** SOAP messages[3] can cross multiple organizational boundaries, each requiring its own authentication process, it is change to have authentication system for each document. This can be done using SAML.

9. **Anomaly Detection:** It is generally used method to detect the abnormal behavior of the[8] web service to detect whether it is an attack or not. It is founded on analyzing parameters associated with SOAP request like size, frequency of equal incoming message.

   Few mitigation techniques that can often base in prevention mechanisms can also decrease the after effects of an attack.

**Mitigation:** web services are standard for applications to transfer information over the internet and also these are the building blocks of most prevalent business architecture today[8] i.e. SOA (Service oriented architecture). Web service relies heavily on the xml, due to the use of SOAP And WSDL[3] and this is the reason that makes them so vulnerable to the xml based attacks. These attacks are important to prevent as if does not, they can cause a huge harm to the entire Services architecture of the Business Enterprise.

![Figure2. Taxonomy on The Counter Measures to attacks on Web service](image)

**Architecture:**

5. **ALGORITHM DESCRIPTION**

A Graphical Schema Matching Algorithm

**Cycle Detection Algorithm:**

Require: Table table
For each pair p(a_i,b_j) in pairs; do
Topsort(pairs);
If ((decedents(a_i) X ancestors(b_j)) ∪ (ancestors (a_i) X deceidents (b_j))) ∩ pairs ≠ empty ) then returntrue;
end if
end for

1. Schema matching is to find semantic correspondences among elements of two schemas[8],[7]. Most of the suggested approaches concentrated on the similarity of separate elements or at most neighborhood information, rather than on the
global semantics of the schemas[3]. We suggest a novel approach to the schema matching problem utilizing global semantics. A schema is represented by an acyclic direct graph, where nodes represent elements or attributes and links represent the containment relationships.

2. Schemas[6] are represented by acyclic graphs, which do not allow containment cycles that create a semantic contradiction. If a semantic between two schemas has no semantic contradiction, we call the mapping harmonic.

3. If a cycle exists in the mapping pairs[8], the algorithm demands to choose a mapping pair to adjust to remove the cycle. Conceptually, the pair that generates or creates the most crossings should be removed, i.e., the key contradiction pair.

4. Then our algorithm breaks the cycle by finding the second most suitable mapping for the element in the table. After breaking cycles, the algorithm creates mapping pairs established on the new similarity table and iteratively finds and breaks new crossings until no more crossing can be found or a threshold is reached. The pseudo code is described in algorithm.

5. Merge with Security Property

6. Having the mapping between two methods, one can merge the two models to create a federation and exchange information. The security extension of merge eases the process by automatically generating access control rules[5] for the output data model.

7. The process of merging two access control rules is called access merge[6]. It is based on subject mappings.

8. ALLOW VS. DENY

Suppose rule 1 allows subject s1 the access to object o1 in model M1 and rule 2 denies[8] the access of s2 to o2, where s1 mapped to s2, o1 mapped to o2. Conflict arises when two subjects and their respective objects are merged, i.e., s1 and s2 into one subject (called S3) and o1 and o2 into one object (O3). Whether to allow the access of S3 to O3 would be a delicate issue. Possible include:

1. Deny the subjects access in the resulting rule;
2. Allow the subjects access in the resulting rule;
3. Separately create two rules for each subject, and remove the mapping between the two subjects;
4. Request a user intervention.

Allowing all the access will break the access control rule[5] for M2. Solution (4) requires user’s intervention and will produce a result depending on the policy. Users intervention requires a user with the security extension for model management to be user friendly, i.e., denies the subjects access and requests users’ intervention, thus provides or supplies safe suggestions that are customizable.

6. EXPERIMENTAL EVOLUTION:

Web services are increasingly being used to provide critical operations in business-to-business and safety-critical environments. In these environments, the exploitation of security vulnerabilities[7] may result in major damages in the services infrastructures, financial or reputation losses to the organizations involved, and other catastrophic consequences for the users and the environment. Web services frameworks are the basis for developers to create and deploy web services, and must provide a robust and secure environment, so that an application can deliver its service, even when in presence of security attacks. In this paper we study the behavior of well-known web services frameworks in the presence of security attacks[8] targeting the core web services specifications, i.e., those enabling basic message exchange functionalities. Results show that frameworks are quite resistant to attacks. However, they also indicate that even very popular and highly tested frameworks can be vulnerable to attacks, with potentially catastrophic consequences for the services being deployed.
7. Graph:

8. Future Work and Conclusion:
The future work relates to signing cookies and hidden fields manipulation attacks. Also, URL patterns will be utilized in describing sites with dynamic resources.

In this paper we introduce XML schema, XML, usage control and discuss access models for XML schemas and XML documents[1] and fine-grained by using usage control[11]. The usage control models supplied an approach for the next generation of access control. In usage control we examined fine grained access control. The access control in web services are in xml format. The web services represent a new and un explored set of security-sensitive technologies that have been widely deployed by large companies, governments, financial institutions and in consumer applications. Web application attacks (SQL injection, XSS, etc.).[8] In our proposed system is based on the anomaly-based methodology it proved to the able to protect web application from both known and unknown attacks. The web services are powerful, easy-to-use, and open. The lots of security work still required the web services developing many analysis they are WS-security, WS-routing and WS everything. The schema based usage control model can be a solution of the Web service security.

9. REFERENCES
8. Abhinav Nath Gupta and Dr. P. Santhi Thilagam., ATTACKS ON WEB SERVICE NEED TO SECURE XML ON WEB, October 2013, 2-10.