International Journal of Security and Its Applications
Vol.9, No.7 (2015), pp.179-188
http://dx.doi.org/10.14257/ijsia.2015.9.7.16

The Study of Access Control Model Using XML

Huanming Zhang [#], Quanlong Guan [#1] and Weiqi Luo

Network and Education Technology Center, 510632, Jinan University, China
[#] These authors are co-first authors, who contributed equally to this work,

Abstract

XML, the Extensible Markup Language, had become an important tool for both storage and exchange of data. As the applied areas of XML had been widen gradually, the security problems of XML became a main concern. Hence, the study of access control using XML had been an important topic of security study of XML nowadays. In this paper, we would first made a brief introduction of access control using XML, and some requirements of XML access control would be included. After that, we would give a detail presentation of an access control model using XML, and point out the most significant feature of it. Finally, we analyzed the direction and difficulty in the study of access control using XML, and then illustrate the practical significance of the study.

Keywords: *XML, authority, access control, security, XML schemas*

1. Introduction

XML was an open standard of data representation, along with lots of advantages: concepts of elements, extensible, separation of display and content, complex structure presentable, and etc. XML had played an important role in almost all E-commerce systems and Web applications. As XML was just a language, security of its application could not be guarantee by itself. Hence, there was a need to build secure application platform as the precondition and foundation of its applications.

Generally, security threats faced by E-commerce with such following classification [1-2]: illegal access, illegal tempering, counterfeiting, repudiation, denial of service. And to force those mentioned secure threats, the E-commerce application system must fulfil the following needs: data confidentiality, access control, authorized identity identification, data integrity, and denial-anti. There some new work about this area. E Damiani *et al*. [9] propose five basic requirements for standardizing XML access control at the tag level. Venkatasubramanian *et al.* [10] proposed an Adaptive and proactive Access Control Approach for Emergencies in Smart Infrastructures. Algarín A *et al*. [11] generated XACML enforcement policies for role-based access control of XML documents. Hao H *et al*. [12] proposed API-level access control in smartphone analysis. N Farooqi applied dynamic trust strategy based access control to Improve XML databases security. Wonohoesodo and Tari [14] proposed access control model using XML for web services.

And these threats could be solved with the use of data encryption and digital signature technique [3-4]. These two techniques co-working with access control to provide a safe platform for both data exchange and distribution using XML. Obviously, an access control system with multi-level and multi-granularity access control ability was an important direction of recent study of XML security problem.

[1]***Corresponding author*** e-mail: gql.jnu@gmail.com

ISSN: 1738-9976 IJSIA

2. Requirements of XML to Access Control

Protection could not be guaranteed while using traditional access control uniquely because of some special features of XML documents, hence there was a need to declare a special access model using XML. By analyzing existing models, we found that those model were all based on the declarations of a group of authorities, and these authorities must at least contain the subject of their application, protected object, and the upcoming execution. Difference of recent XML access control models were mainly presented by different implementation of the access to subject, and object.

We would discuss the basic features of existing access control model[5] using XML:

Better Access Control Granularity: Access control would support different levels of granularity of access control, from documents set, single document, elements set, single element, to particular elementary content.

Support Levels of Authority [6]: In many instances of access control, mono-concentration authority could not adopt to the multi-level request from application environment. Good access control system should support global and local (or even more levels) authority.

Support for Web Technique: With the use of Web site, XML document would always usable. Without using existing API and development tool, XML access control should be convenient and Web technique integrated.

Transparency: Access control operation should be as transparent as possible to the requester. Requester should not able to notify message hidden by access control system in a document. Moreover, access control should guarantee the effectiveness of document to its DTD

Good Compatibility and Interoperability [7]: Access control should conveniently interoperate with other system

Integration with Existing User Authorization Technique: Access control should easily integrate other user authorization technique.

3. Access Control Model using XML

The following would be an introduction of an access control model using XML, access control of XML schemas or instance documents need the declaration of subjects and objects, and the access control rules especially for subjects and objects.

3.1 Subject

The xml subject which mean a user or a group of users. Each user had a notification symbol, which could be used as the user login name also. Each user or user group described by user features document. Then safety administrator could define the system safety rules according to the user features document. The XML document shown in Figure 3.1 was a simple user characteristic document.


```
<userProfile>
  <users>
    <user id="tang" name="Tang" />
    <user id="ma" name="Ma" />
    <user id="liu" name="Liu" />
    <user id="huang" name="Huang" />
    <user id="lee" name="Lee" />
  </users>
  <groups>
    <group id="professor" >
      <member uidref="tang" />
      <member uidref="huang" />
    </group>
    <group id="researchAssistant" >
      <member uidref="liu" />
      <member uidref="lee" />
    </group>
    <group id="financial">
      <member uidref="ma" />
    </group>
  </groups>
</userProfile>
```

Figure 3.1. User Features Document in XML Format (UserProfile.xml)

For simplifying definition of authority, some access control model allowed the authority specification defining subject into following three categories:

User Group: statically defined a group of user, which could be nested and overlapped

Position Mode: a group of position set, acquired by adding the character * in front of the physical or symbolic address

Role: according to authority set, users could determine their role dynamically.

3.2 The Protected Object

The protected object, could be documents, outlines or part of any of them. We could appoint security strategy to following object:

All instance of a certain given schema

Well-formed XML document set

Some parts of one or multiple document

The granularity of protected object could be controlled based on the access control of content, hence security administrator could confirm security rules based on document content (attribute value or element value). The effect of this mechanism was tremendous, as we would have to set different protection need based on different content as the structure of document was usually the same. Figure 3.2 showed us an example of XML document, which describing a set of some high school research project.

```
<ResearchProjects>
  <project prjID="201207031" type="public">
    <title>XML Query</title>
    <funding>
      <organization>NSFC</organization>
      <amount>100000</amount>
    </funding>
    <principalInvestigator>
      <name>xxx</name>
      <email>xxx@jnu.edu.cn</email>
    </principalInvestigator>
    <patent pID="2002060504">
      <title>Simple XML QueryModel</title>
      <abstract>.....</abstract>
    </patent>
    <paper>
      <title>A XML QueryModel</title>
      <author>Tang</author>
      <abstract>.....</abstract>
    </paper>
  </project>
  <project prjID="201423313" type="confidential">
    <title>XML Database Security</title>
    <funding>
      <organization>EC Company</organization>
      <amount>2120100</amount>
    </funding>
    <principalInvestigator>
      <name>lsh</name>
      <email>lsh@jnu.edu.cn</email>
    </principalInvestigator>
    <patent pID="2013307323">
      .....
    </patent>
    <paper>
      .....
    </paper>
  </project>
</ResearchProjects>
```

Figure 3.2 XML Document Project (Research.xml)

3.3 Operation Permission

Most of the recent XML access control model supported read operation only. That was because there was not yet a standard for the language in any XML update. Management of write right was difficult, access control strategy and DTD definition for XML document (or XML Schema) need to be considered at the same time. In fact, as some elements or attributes might be defined as deny by the access control strategy, therefor, DTD might be partly hidden while accessing XML document of user. For example, when adding an element to a document, the user might not actually know the existence of related, integrant, attributive element, as user had no authority to access the attribute itself.

However, some method had tried to support the authority of write operation; write operation classified as followed: insert, update, and delete.

In the work, Gabillon *et al.* [8] proposed that classified read authority into two classed: authority to read one element content, and authority to know some position of an element in XML document but not knowing its name and its content. The former would be transformed into read operation by the model; the latter would be transformed as positioning operation. Meanwhile, author proposed providing choice with authority and transmission of authority to security administrator in classical database environment.

3.4 Security Rule

The security rule is a 7-tuple group in following format:
<subject, target, path, privilege, action, propagation, priority>

subject, which registered user or user group, special user “all” represent all user

target, the XML document going to be accessed

path, used to specify a node in XML document tree, could be confirmed by XPath. Find the protected object out using the XML document pointed by target and the node pointed by path.

privilege, an element from set {grant, deny}

action, an element from set {read, write}, defining all :: = read || write

propagation, an element from set {NO_PROP, FIRST_LEVEL CASCADE}

priority, optional, default priority was 0

The authorizing process was actually the process of security administrator setting safety rules. XML document shown in Figure 3.3 was an example of security rule base, which was describing the authority to the document shown in Figure 3.2 for user mentioned in Figure 3.1.

```
<securityRules>

   <rule subject= " researchAssistant" target= "ResearchProjects.xsd" path= "/ResearchProjects/project/paper" action= " read" privilege= "grant" propagation= "FIRST_LEVEL" priority= 0>

   <rule subject= " researchAssistant" target= "ResearchP ro jects. xsd" path= "/ResearchProjects/project[@type= 뭔onfidential?]" action= " read" privilege= "deny" propagation= "CA SCADE" priority= 1>

   <rule subject= " financial" target= "ResearchProjects. xsd" path= "/R esearchProjects/project/funding/amount" action= " read" privilege= "grant" p ropagation= "NO _ PROP" priority= 0>

   <rule subject= "professor" target= "ResearchProjects. xsd" path= "/ ResearchProjects/project/patent" action= " read" privilege="grant" propagation= "CA SCADE" priority= 0>

   <rule subject= "all" target= "ResearchProjects. xsd" path= "/" action= "all" privilege= "deny" propagation= "CA SCADE" priority=-10>

</securityRules>
```

Figure 3.3. Safety Rule Base (AccessRules.xml)

Authority setting shown in Figure 3.3 were as follow:

The first and second rules set members in group “researchAssistant” able to read project paper in reading type “public”. This user group had no authority to read project information with type “confidential”

The third rule set members in group “financial” able to read all “funding amount” elements in the project

The forth rule set members in group “professor” able to read all patent information in the project

The fifth rule define for all user, if authority “grant” was not be given directly, there was no right to access any instance of schema ResearchProjects.xsd

3.5 Authorization Collision and Analysing Strategy

Authority set by safety rules could be classified into direct authorization and indirect authorization.

Direct authorization meant that an explicit authorization was given to the user target object with one or some safety rules. For instance, assumed user “tang” willing to access all content pointed by “/ResearchProjects/project/patent” in the document Research.xml, the forth rule in Figure 3.3: < rule subject = " professor" target = "ResearchProjects. xsd"

path= "/ResearchProjects/project/patent" action= " read" privilege= "grant" propagation= "CA SCADE" priority= 0> was a direct authorization to this request.

Indirect authorization was for the user aiming target; as there was no specific safety rule for the target in the safety rule base, target could acquire safety authority indirectly by the transmission of other rules. For example, there was such a rule like Figure 3. 4:

```
<rule subject = "tang" target = "Research. xml" path= "/ResearchProjects/project " action= "
read" privilege= "grant" propagat ion= "CA SCADE" priority= 0>
```

Figure 3.4 An Example for Authorization Rule Control Strategy

If user “lsh” would like to access all of the content pointed by “/ResearchProjects/project/paper” in document Research.xml, then through transmission method “CA SCADE”, the safety rule authorized user “tang” with right “ read” to target path “/ResearchProjects/project/paper” and its sub-tree. This was a kind of indirect authorization.

Authorizing “grant” and “deny” simultaneously may potentially cause authorization collision, because a user defined same authority to one protected object, and the only difference is the accessing mode. The kind of authorization collision might cause directly or indirectly. Direct collision meant there were two safety rule authorized “grant” and “deny” respectively; indirect collision was the collision caused by transmission. This paper allowed this kind of collision, way to solve collision was to declare a kind of collision analysing strategy based on the highest priority principle, as follow:

To node i and user u, if there was a group of rule that with collision, then chose the one with higher priority

If there were more than one rules had been chosen, then chose the first rule in that group only

3.6 Features of Our Model

Our access control models using XML mentioned above were with following features:

1. Support Authorization with Fine and Coarse Granularity: That model support model-level and instance-level authorization; model-level authorization was to fulfil a certain authorization to all instance of DTD; instance-level authorization was to authorized a certain particular XML document, where the authorization could be refined to certain part, certain element or attribute of the document

2. Take Two Different Transmission Strategy: transmission strategy of authorization basically separated into local and recursive; local meant the authorization of certain element was only applied to all attributes; recursive meant the authorization of certain element would be applied to its attributes and sub-elements. Generally, “grand” could be “local” or “recursive” authorization, and “deny” would be “recursive” authorization.

3. Provide Support to Abnormal Condition: XML access control model was facing two kinds of abnormal condition: authorization collision, and incompletion problem. We could use the higher priority principle to authorization collision; and for incompletion problem, as “grand ” and “deny” had not be clearly invested, a “open” or “close” strategy would be used by default generally, which meant accept “grand” or “deny” authority.

4. Evaluation

In this paper, our experiments using SUNXACML, JDOM parser and Java language. SUNXACML developed by Sun Java-based XACML API, which provides the PDP and

PEP implementation. The simulation is carried out in the following environments: Intel Core i3-2120 3.3GHZ CPU processor, 8GB of memory, Windows 7 operating system. JDOM version is 2.0.5 and Java is SE 7.0 (1.7.0). Testing with DTD and XML documents by XMark [16] obtained authorization rules set XPath formula based on the DTD is generated by YFilter [17] The XPath tool, query the artificial setting.

The proposed model are proof of concept level, the benchmark index is used to obtain the differences in performance. Since our sample XML data is generally small, and provide experimental file type size were 2KB, 5KB, 8KB, 10KB, 15KB, 20KB, not likely to reflect the efficiency of the model, the experimental results produced are not comparable. To properly measure the results, we take 1024 iterations of experimental data obtained 2M, 5M, 8M, 10M, the amount of data 15M, 20M of. Disposable minimize overhead, XML input completed 1024 iterative resolution files of different sizes. Benchmarking tests were repeated six different types of files, to see how to deal with six different models in size. This means that the first iteration of six different sizes of file 1024, and then measure. Before parsing and calculation, the file is completely loaded into memory. This does not include the loading time consuming to resolve. 1024 iterations estimates are carried out in its own process. This means that, for each file in a separate process for resolution. A process running again. Each file is estimated that three times. 1000 file parsing process start and stop three times, the final calculation of the average value of their time. Processes are performed sequentially, not in parallel.

In addition, set the following parameters are used to measure the inquiry, parser and access control time:

- PT_XML: The parsing time to all the XML documents.
- PT_RULE: The parsing time to authorized rules for accessible data.
- QT_XML: The total query time to all the XML documents.
- QT_DTD: The query time for all the DTD.
- QT_DATA: The total query time to accessible data.
- QT_EL: The query time to some xml elements.

1024 iterations estimates are carried out in its own process. This means that, for each file in a separate process for resolution. A process running again. Each file is estimated that three times. 1000 file parsing process start and stop three times, the final calculation of the average value of their time. Processes are performed sequentially, not in parallel. The time of iteration result is shown in table 3.1.

Table 3.1 Time Evaluation for Authorization Rule Control in xml Documents

	PT_XML	PT_RULE	QT_XML	QT_DTD	QT_DATA	QT_EL
Time_2M	11.362	13.067	28.049	16.336	22.317	28.372
Time_5M	18.259	14.127	41.762	17.716	26.633	32.918
Time_8M	21.123	15.032	43.025	18.313	32.483	35.742
Time_10M	27.308	17.232	58.235	22.409	36.847	38.007
Time_15M	35.328	22.928	76.565	24.274	49.145	40.351
Time_20M	49.853	23.504	84.911	25.302	54.265	42.981

We can see from Table 3.1, the file size increases, xml document after access control, processing time increases. The result is in line with our expectations. We carefully observe these data, the first column indicates the parsing time. 2M files only 11ms, but after the file to 20M, corresponding parsing time increased to 49ms. The parsing time for access control rule, but the gap is relatively small and the change from 13ms to 23ms, just adds 10ms. From this point of view after the explanation, increasing access control, will not result in the document xml parser and query operations such as the impact is very small. The third column of data is time for all queries xml document parser time longer than that due to Find, compare the results from the data, it is easy to see the lateral. For example, 8M of data, PT_XML only 21ms, 43ms and QT_XML reached the same token,

xml document 20M's, PT_xml time is only 49. And QT_XML time to reach 84ms. Then look at the fourth column, the query time DTD QT_DTD. Due to the structure of the document DTD is relatively simple, and does not need to repeat the query, so it takes almost the same time, from 16ms to 25ms. The fifth column is accessible data query time, we are using the same benchmark, to find some of the same data The results obtained were compared. With the increase in file size, so the query time spent increases. The last one is the query xml elements in some time, and QT_DATA very similar. General terms, we can see that our proposed access control model to reduce the original xml document processing efficiency, will not cause much impact, is manageable.

5. Conclusions

By its advantages, XML was becoming a general media for data exchange and representation, widely used in E-commerce, became the core while constructing Web Services. These application domain required some safety requirements for certain level, but XML was just a kind of markup language which was not able to guarantee the safety for those applications using XML as their base. Design and implementation of access control model using XML would like to be an important tool to guarantee the application safety. Recently, there were three directions for the study of XML access control:

1. Access control method based on XML: ACT (access condition table), SMT (strategy matching tree), static analysis and etc.
2. Access control strategy language based on XML: Recently, some expressive and functional strategic standard language like XACL, XACML and WS-Policy had been designed to implement standard manual of safety strategy using XML. As semantics and syntactic of those language were still complex, hence, developed an expressive XML access control strategy language with easy semantics and syntactic was an important direction for the recent study of XML access control
3. Design of XML access control model: in recent years, kinds of access control models had been proposed, like model based on safety view or fine granularity.

Although some achievement had been gained from the study of XML access control, like some XML access control method and model, study was still facing lots of difficulty:

1. How to develop suitable method for XML access control based on traditional access control method
2. Where to find an expressive access control strategy language with simple semantics and syntactic; as most of recent XML access control strategy language were syntactic-and semantics difficult, which made the standard manual, integration, management, and maintenance of strategy difficult
3. How to integrate XML encryption and signature with access control effectively; and the way of data model transformation, in order to provide a safe platform for E-commerce and E-government.

The greatest advantage for safety model using XML was that the safety granularity had been refined to elements and attributes level of XML document, different safety strategies could be applied to different parts of the same document. Applied safety process to encryption and digital signature, saving network resource. Meanwhile, different vision to different user to the same documentation could be implemented, user would just able to see authorized content; this function was valuable in no matter E-commerce, E-government, or management in enterprise and government.

Acknowledgements

Huanming Zhang and Quanlong Guan are co-first authors, who contributed equally to this work. This work was supported by Major Program of National Natural Science Foundation of China(No.61133014, No.61272415, No.61272413), the CEEUSRO project of Guangdong province,China (No.2012A080102007, No.2011B090400324,

No.2010A011200038, No.2012B040305008), Science and Technology Planning Project of Guangzhou city(No.11A12070544, No.2013Y2-00071),the Project for Engineering Research Center of Guangdong Province(No.GCZX-A1103), and the Fundamental Research Funds for the Central Universities(No. 21613336).

References

[1] Y.-s. Liu, H. Zhong and Y. Wang, "XML Access Control Model and its Application", Journal of Chinese Computer Systems, vol. 5, (**2005**).
[2] F. Feng, X. Zhou, X. Feng and B. Ye, "The XML fine-grained access control model based on priority [J]", Journal of Computer Applications, (**2006)**, (S2).
[3] S.-h. Tang, "Methods on XML Authorization and Access Control [J]", MINI- MICRO SYSTEMS, vol. 3, (**2005).**
[4] Z.-m. Wang and D.-w. Cui, "Access control strategy based on RBAC for XML security [J]", Computer Engineering and Applications, vol. 43, no. 17, (**2007**).
[5] L. Li, Y.-Z. He and D.-G. Feng, "A Fine-Grained Mandatory Access Control Model for XML Documents [J]", Journal of Software, vol. 10, (**2004**).
[6] H. Fu, H. Li and Y.wang, "An Overview of XML and XML-Related Security [J]", Application Research of Computers, vol. 2, (**2004**).
[7] S. De Capitani di Vimercati1, S. Foresti1 and S. Paraboschi2, "Access Control Models for XML", http://www.springerlink.com/content/j8353g682562qk1k/fulltext.pdf.2007,12
[8] A. Gabillon, "An authorization model for XML databases", In: Proc. of the 2004 Workshop on SecureWeb Service (SWS04), Fairfax, Virginia (**2004**) November.
[9] E. Damiani, De Capitani di Vimercati S and S. Paraboschi, "A fine-grained access control system for XML documents [J]", ACM Transactions on Information and System Security (TISSEC), vol. 5, no. 2, (**2002**), pp. 169-202.
[10] K. K. Venkatasubramanian, T. Mukherjee and S K S. Gupta, "CAAC--An Adaptive and Proactive Access Control Approach for Emergencies in Smart Infrastructures [J]", ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 8, no. 4, (**2014**), p. 20.
[11] A. De la Rosa Algarín, T B. Ziminski and S A. Demurjian, "Generating XACML Enforcement Policies for Role-Based Access Control of XML Documents [M]//Web Information Systems and Technologies. Springer Berlin Heidelberg, (**2014**), pp. 21-36.
[12] H. Hao, V. Singh and W. Du, "On the effectiveness of API-level access control using bytecode rewriting in Android [C]", Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, ACM, (**2013**), pp. 25-36.
[13] N. Farooqi, "Applying Dynamic Trust Based Access Control to Improve XML Databases Security [J]", (**2013**).
[14] R. Wonohoesodo and Z. Tari, "A role based access control for web services[C]", Services Computing, 2004, (SCC 2004), Proceedings, 2004 IEEE International Conference on, IEEE, (**2004**), pp. 49-56.
[15] A. Schmidt and F. Waas, "XMark: a Benchmark for XML Data Management [C]", Proceedings of the 28th VLDB Conference, China, (**2002**).
[16] Y Filter 1.0 Code Release [EB/OL]. [2005-4]. (http://yfilter.cs.berkeley.edu/code-release.htm)
[17] E. Damiani, S. Vimercati, S. Paraboaschi and P. Samarati, "Design and implementation of an access processor for xml documents", In Proceedings of the 9th international WWW conference, Amsterdam, (**2000**), May.
[18] E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati, "Securing xml document", In Proceedings of the 2000 International Conference on Extending Database Technology (EDBT2000), pp 121-135, Konstan, Germany, (**2000**), March.

Authors

Huanming Zhang, Current position, grades: Senior Engineer, Network and education technology center, Jinan University (China). His research interests include Internet and network.

Quanlong Guan, Current position, grades: Associate Professor, Network and education technology center, Jinan University (China). His research interests includes Smartphone security, Cloud Security, Internet of Things and Mobile Security.

Weiqi Luo, Current position, grades: Porfessor, Network and education technology center, Jinan University (China).